



**University of  
Zurich**<sup>UZH</sup>

**Zurich Open Repository and  
Archive**

University of Zurich  
University Library  
Strickhofstrasse 39  
CH-8057 Zurich  
[www.zora.uzh.ch](http://www.zora.uzh.ch)

---

Year: 2014

---

## Dynamic alignment models for neural coding

Kollmorgen, Sepp ; Hahnloser, Richard H R

**Abstract:** Recently, there have been remarkable advances in modeling the relationships between the sensory environment, neuronal responses, and behavior. However, most models cannot encompass variable stimulus-response relationships such as varying response latencies and state or context dependence of the neural code. Here, we consider response modeling as a dynamic alignment problem and model stimulus and response jointly by a mixed pair hidden Markov model (MPH). In MPHs, multiple stimulus-response relationships (e.g., receptive fields) are represented by different states or groups of states in a Markov chain. Each stimulus-response relationship features temporal flexibility, allowing modeling of variable response latencies, including noisy ones. We derive algorithms for learning of MPH parameters and for inference of spike response probabilities. We show that some linear-nonlinear Poisson cascade (LNP) models are a special case of MPHs. We demonstrate the efficiency and usefulness of MPHs in simulations of both jittered and switching spike responses to white noise and natural stimuli. Furthermore, we apply MPHs to extracellular single and multi-unit data recorded in cortical brain areas of singing birds to showcase a novel method for estimating response lag distributions. MPHs allow simultaneous estimation of receptive fields, latency statistics, and hidden state dynamics and so can help to uncover complex stimulus response relationships that are subject to variable timing and involve diverse neural codes.

DOI: <https://doi.org/10.1371/journal.pcbi.1003508>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-107800>

Journal Article

Published Version



The following work is licensed under a Creative Commons: Attribution 4.0 International (CC BY 4.0) License.

Originally published at:

Kollmorgen, Sepp; Hahnloser, Richard H R (2014). Dynamic alignment models for neural coding. PLoS Computational Biology, 10(3):e1003508.

DOI: <https://doi.org/10.1371/journal.pcbi.1003508>

# Dynamic Alignment Models for Neural Coding

Sepp Kollmorgen\*, Richard H. R. Hahnloser

Institute of Neuroinformatics, University of Zurich/ETH Zurich, Zurich, Switzerland

## Abstract

Recently, there have been remarkable advances in modeling the relationships between the sensory environment, neuronal responses, and behavior. However, most models cannot encompass variable stimulus-response relationships such as varying response latencies and state or context dependence of the neural code. Here, we consider response modeling as a dynamic alignment problem and model stimulus and response jointly by a mixed pair hidden Markov model (MPH). In MPHs, multiple stimulus-response relationships (e.g., receptive fields) are represented by different states or groups of states in a Markov chain. Each stimulus-response relationship features temporal flexibility, allowing modeling of variable response latencies, including noisy ones. We derive algorithms for learning of MPH parameters and for inference of spike response probabilities. We show that some linear-nonlinear Poisson cascade (LNP) models are a special case of MPHs. We demonstrate the efficiency and usefulness of MPHs in simulations of both jittered and switching spike responses to white noise and natural stimuli. Furthermore, we apply MPHs to extracellular single and multi-unit data recorded in cortical brain areas of singing birds to showcase a novel method for estimating response lag distributions. MPHs allow simultaneous estimation of receptive fields, latency statistics, and hidden state dynamics and so can help to uncover complex stimulus response relationships that are subject to variable timing and involve diverse neural codes.

**Citation:** Kollmorgen S, Hahnloser RHR (2014) Dynamic Alignment Models for Neural Coding. PLoS Comput Biol 10(3): e1003508. doi:10.1371/journal.pcbi.1003508

**Editor:** Olaf Sporns, Indiana University, United States of America

**Received:** March 3, 2013; **Accepted:** January 28, 2014; **Published:** March 13, 2014

**Copyright:** © 2014 Kollmorgen, Hahnloser. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Funding:** This work was funded by the Swiss National Science Foundation (grant 31003A\_127024) and by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007–2013/ERC Grant AdG 268911). The funders had no role in study design, data collection and analysis, decision to publish, or preparation of the manuscript.

**Competing Interests:** The authors have declared that no competing interests exist.

\* E-mail: skollmor@ini.phys.ethz.ch

## Introduction

Neural response models are used to relate neural activity to sensory stimuli and motor behavior. A very common type of neural response model is comprised of a linear stage, at which one or more linear filters (often referred to as receptive fields) are applied to the stimulus, and a subsequent non-linear stage that converts the filter outputs into a spiking probability that feeds into a Poisson process generating the spikes [1]. More precisely, the spiking probability (i.e., the instantaneous rate of the Poisson process) of a neuron is modeled as  $P(\text{spike}|S) = f(BS)$ , where the column vector  $S$  represents the stimulus,  $f$  the nonlinearity, and  $B$  is a row vector containing the linear filter or a matrix in case of several filters. Variations of these linear-nonlinear Poisson cascade models (*LNP models*) have been studied extensively [2–5]. Parameter estimation techniques range from spike triggered averaging in case of one linear filter and white noise stimuli, to spike triggered covariance [1,6] in case of several linear filters, and maximally informative dimensions, in case of one or several linear filters and no restrictions on the distribution of stimuli [2,4]. Although these techniques are effective in many domains, they fail in others, where the neural code might be more intricate (detailed below).

A crucial assumption about the relationship between stimulus and response inherent in these techniques is that the response latency of the cell, the filters (or receptive fields), and the non-linearity all are assumed to be the same throughout the experiment (Fig. 1A). This assumption of a *fixed stimulus-response relationship* is, however, not necessarily valid. On the one hand, the relationship

between stimulus and response, the neural code, could vary in time (Fig. 1C). On the other hand, the response latency could be noisy or vary systematically (Fig. 1A).

The extent to which a *fixed stimulus-response relationship* applies to neurophysiological data is unclear. In terms of spike timing, a fixed relationship entails both constant response latency and an amount of spike-time-jitter that is smaller than the relevant temporal structure of the receptive field. However, noisy response latencies (Fig. 1A) are observed in almost all electrophysiological studies because neural systems are intrinsically noisy. Variations in response latency to a repeated stimulus (measured as the standard deviation of time of first spike after stimulus onset) in the range of 3–5 ms have been reported already at a very low stage of the visual system, in retinal ganglion cells [7]. Variability in response latency can be notably larger in cortical areas. For instance, variations in first-spike latency (again measured as the standard deviation) of up to 12.5 ms have been observed in single cells in ferret primary visual cortex in response to flashed natural images [8]. Furthermore, systematically varying response latencies have been demonstrated in various model systems, e.g., image contrast modulates response latency both in retinal ganglion cells [9] and in visual cortical neurons [10–12], fueling discussions about the role of spike latency in neural coding [12,13]. In a recent study, latencies of cells in macaque inferotemporal cortex were found to systematically differ for primate and non-primate face stimuli, with latency differences on the order of tens of milliseconds [14].

When latencies are strongly fluctuating or spike time-jitter is large, many modeling techniques that assume a *fixed stimulus-response relationship*, such as spike triggered averaging, yield

## Author Summary

The brain computes using electrical discharges of nerve cells, so called spikes. Specific sensory stimuli, for instance, tones, often lead to specific spiking patterns. The same is true for behavior: specific motor actions are generated by specific spiking patterns. The relationship between neural activity and stimuli or motor actions can be difficult to infer, because of dynamic dependencies and hidden nonlinearities. For instance, in a freely behaving animal a neuron could exhibit variable levels of sensory and motor involvements depending on the state of the animal and on current motor plans—a situation that cannot be accounted for by many existing models. Here we present a new type of model that is specifically designed to cope with such changing regularities. We outline the mathematical framework and show, through computer simulations and application to recorded neural data, how MPHs can advance our understanding of stimulus-response relationships.

suboptimal results [15,16]: The estimated receptive fields are blurred and the accuracy of predicted responses to novel stimuli is low (Fig. 1B).

*Fixed stimulus-response relationships* can also be violated in case of changes in intrinsic or hidden brain states [17,18]. For example, neurons in the primary somatosensory cortex of the rat undergo up and down states given by two separate membrane potentials. Spiking responses to whisker deflections in these cells are dependent on whether neurons are in the up or the down state: in the down state, a reliable response is observed, whereas in the up state activity is largely stimulus independent [19]. Stimulus context can also induce changes in internal states. For instance, in an awake marmoset study of single-unit responses in the auditory cortex to sequences of 2 sound stimuli, responses to the second stimulus were not static but depended strongly on the first stimulus. This modulation in second-stimulus responses can last longer than 1.5 seconds [20]. We illustrate dependence of neural computation on intrinsic states in a cartoon (Fig. 1C) in which the simulated neuron switches between 2 static receptive fields. Response switching has also been observed in the amphibian retina. Ganglion-cell activity is typically dominated by OFF responses. However, a large peripheral image shift (as occurs during head saccades) can induce a switch (for a few hundred milliseconds) from transmitting OFF signals to transmitting ON signals [21]. One of the most compelling examples of response switching has been observed in songbirds: many neurons in cortical motor and auditory areas are responsive to playback of sound stimuli except when birds are singing, at which times responses are locked to the song but not influenced by sound playback [22,23]. Hence, if such neural responses were to be modeled across singing and non-singing states, anything but a two-state model would be inadequate. Indeed, many classical models fail in cases of response switching: the estimated classical receptive fields contain superimposed structures derived from the switched responses, which yields sub-optimal results (Fig. 1D).

To address variable response latencies and dynamic neural codes we consider the problem of neural response modeling as an alignment problem. We introduce mixed pair hidden Markov models (MPHs) as novel neural response models allowing for dynamic alignment of stimulus and response without *fixed stimulus-response assumptions*. In the case of varying spike latencies (e.g. when a neuron fires in response to a particular stimulus but with a variable latency or lag, Fig. 1A, B), MPHs help to detect

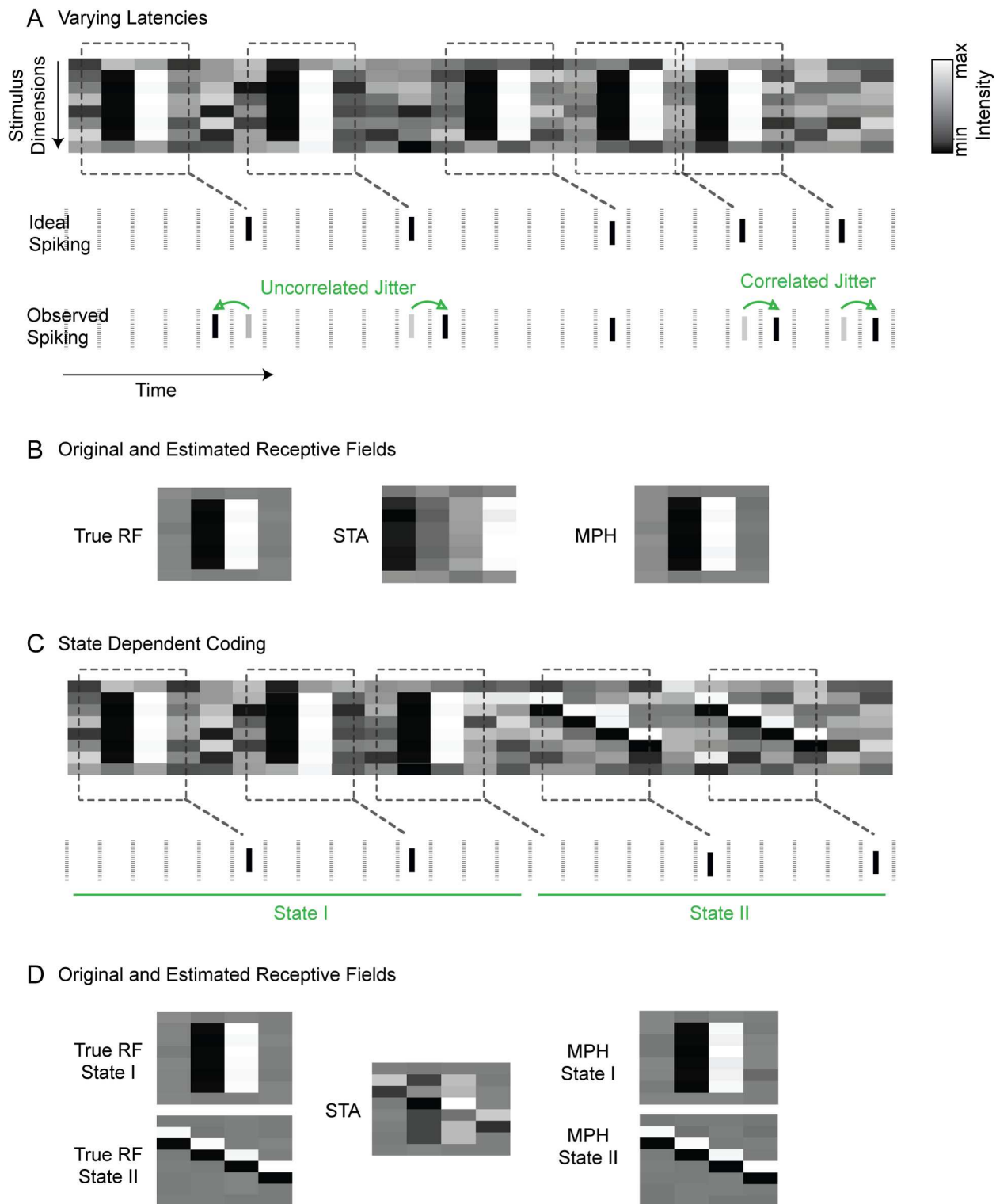
corresponding stimulus-response parts by associating individual spikes with particular stimulus time points; and, they help to uncover stimulus-response relationships including the spike-jitter statistics and the receptive field of the neuron. In case of switching dynamics (e.g., when a neuron switches between being responsive to either one stimulus or another depending on the behavioral state of the animal or a cueing stimulus, Fig. 1C), MPHs help to identify parameters such as the receptive-field switching probabilities and the switching events. Our MPH approach to dynamic alignment combines response switching (*context dependency*) and spike-time jitter or systematically varying latencies (*flexible timing*) in one unified framework. We show how to use stimuli and neural responses to jointly estimate all model parameters including spike time jitter, systematically varying latencies, and switching probabilities. We demonstrate the benefits of dynamic alignment on simulated data and on extracellular data recorded in cortical brain areas of singing birds.

## Results

### Mixed Pair Hidden Markov Models

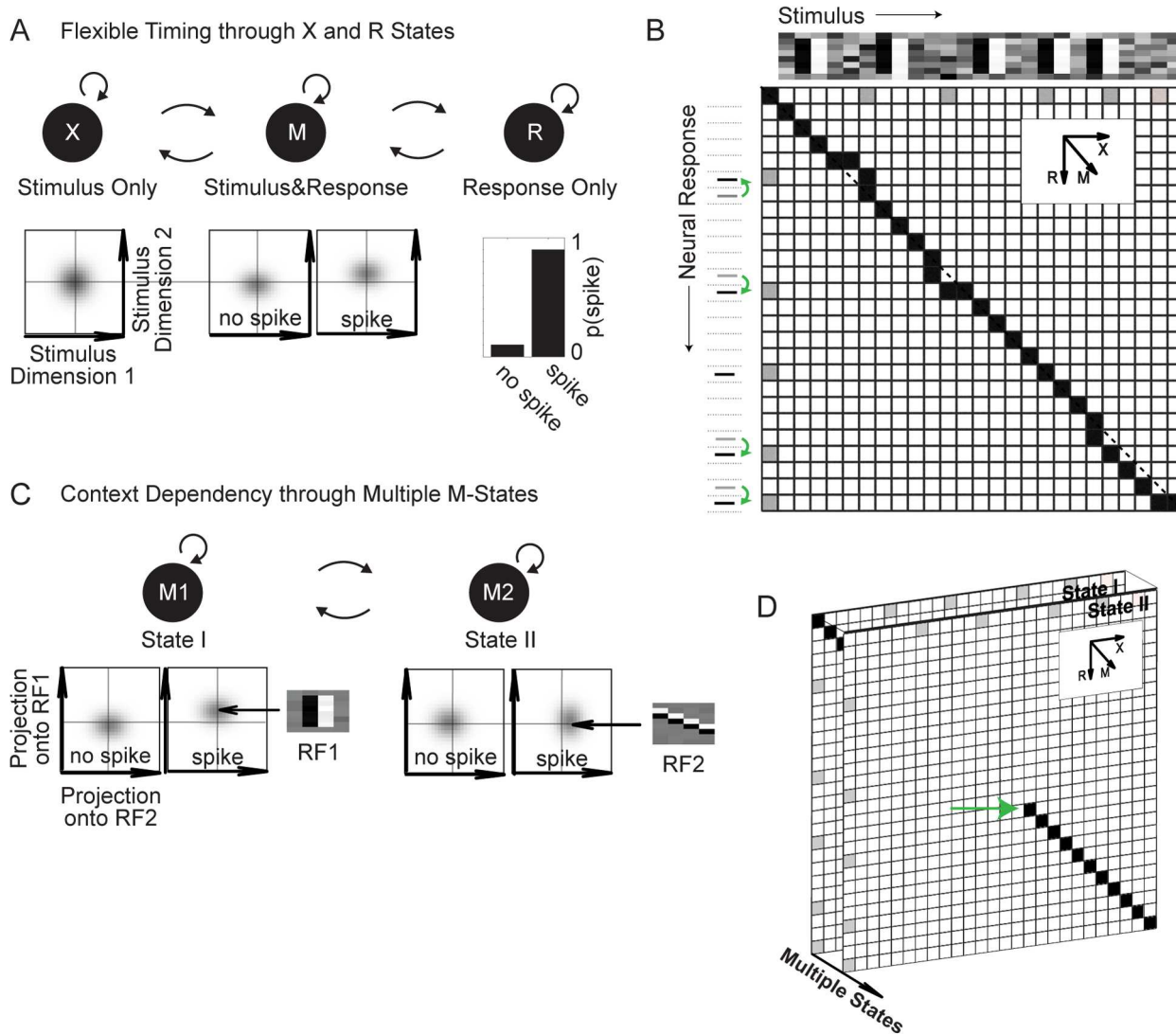
We solve the alignment problem by jointly modeling stimulus and response by a mixed pair hidden Markov model (MPH), which is a generative model for both the stimulus and neural response. In MPHs, different neural codes, i.e. different relationships between neural activity and sensory input, can coexist as different states or groups of states in a Markov chain. MPHs are unlike classical hidden Markov models because they dynamically operate on pairs of sequences - neural activity and stimulus - instead of single sequences (i.e. a joint sequence of neural activity and stimulus). For a mathematically detailed introduction to MPHs (and an introduction to HMMs), see the Materials and Methods section.

We explain the workings of MPHs in intuitive terms by considering first the special case of jittered spike times (Fig. 1A). We assume spikes are associated with the stimulus (i.e. a time window of the stimulus [1]) that precede the spikes by an average time lag  $dT$ . Instead of associating spikes and stimuli at a constant lag  $dT$  (such as is the case for standard spike triggered methods including STA, STC, maximally informative dimensions, etc.), MPHs associate an individual spike with a stimulus at the individual lag  $dT + e$  (in units of stimulus-response bins,  $e$  being an integer,  $e$  can be different for each spike). MPHs achieve this flexibility via three different types of hidden states and by keeping track of the momentary lag  $dT + e$  and its evolution. First, *matching states* (*M-states*) associate a spike with a stimulus at the current lag  $dT + e$  by modeling the joint probability distribution of spike and stimulus (Fig. 2A, middle). The simplest case are Gaussian stimulus models comprising two Gaussians, one of which models stimuli jointly occurring with spikes and the other models stimuli not occurring with spikes (Fig. 2A, middle). A model with only a single such M-state is bound to a fixed lag  $dT$  and is equivalent to an LNP model (under appropriate parameter constraints, see also the section on LNP equivalence below). To achieve a variable lag, we introduce two more types of states: X-states (X stands for the stimulus sequence) and R-states (R stands for response sequence). These states can change the momentary lag  $dT + e$  as follows. An X-state models the stimulus (but not the response) via some probability distribution, for instance a Gaussian (Fig. 2A, left). The X-state changes the current lag from  $dT + e$  to  $dT + e - 1$ . Analogously, an R-state models only the spiking response (but not the stimulus) via a discrete probability distribution (e.g. spike or no-spike, Fig. 2A, right). An R-state changes the current lag from  $dT + e$  to  $dT + e + 1$ . An MPH consisting of an M, X, and R state



**Figure 1. Varying response latencies and context dependent neural coding.** (A) *Varying latencies.* Sequence of 8 dimensional white noise stimuli (e.g. successive frames on a one dimensional screen with 8 pixels). An LNP model generates spikes (black bars) if a chunk of stimulus (dashed rectangles) is similar enough to its receptive field (dashed rectangles). Jitter-free or ideal spikes (vertical black bars, 'ideal spiking') are produced with some fixed latency (dashed diagonal lines). Jittered spikes (black bars, 'observed spiking') are produced by randomly jittering ideal spikes (gray bars) forward or backward in time (green arrows). The jitter of adjacent spikes can be independent or correlated. The jittered spikes are the basis for fitting neural response models. (B) Receptive field (RF) estimates using spike triggered stimulus averaging (STA) on jittered spikes (true RF), jittered spikes (STA), and the MPH on jittered spikes (MPH). Noisy response latencies lead to blurring of STA RFs, but not of MPH RFs. (C) *State-dependent coding.* For the same white noise stimulus, spikes are generated from one of two LNP models depending on hidden states I and II (green lines) determining which model is used. (D) The true RFs are superimposed when estimated with STA. A two-states MPH can faithfully recover the two RFs.

doi:10.1371/journal.pcbi.1003508.g001



**Figure 2. Two minimal MPHs for flexible timing and context dependent coding.** (A) Architecture of the minimal MPH that allows for neural codes with varying latencies, i.e. flexible timing. This MPH has 3 hidden states, one X-state that models only the stimulus, one R-state that models only the neural response, and one M-state that jointly models stimulus and response. The probability distributions over stimuli (bottom) are illustrated as low dimensional projections (stimulus dimension 2 coincides with the receptive field of the M-state). (B) Hidden state sequences in that model correspond to paths in the alignment matrix: a diagonal step leading into position  $(t, u)$  implies that stimulus and response at times  $t$  and  $u$  are jointly modeled by an M-state, a horizontal step implies modeling of only the stimulus at time  $t$ , and a vertical step implies modeling of only the response at time  $u$  (deviations from the diagonal reflect jittered spikes detected by the model). Depicted stimulus and spiking responses are from figure 1A. (C) The minimal MPH for modeling state-dependent neural codes. The MPH can switch between several M-states, each of which represents a different RF. The (projected) stimulus distributions given a spike (spike triggered stimulus ensemble) are centered on the respective RFs (indicated by black arrows). (D) Adding states to the model turns the alignment matrix into an alignment tensor composed of several planes (strictly speaking, B depicts a tensor as well; we just projected all the states onto one plane). The switch from state 1 to State 2 is indicated (green arrow). doi:10.1371/journal.pcbi.1003508.g002

can thus model a spike-stimulus pair via the M state or via the X and the R state. In general fewer states are preferred as there is a cost associated with switching from one state to another. Thus, an MPH will try to keep the lag between stimulus and spike constant unless there is evidence for changing the lag via X and R states.

Intuitively, one can think of an MPH as a finite state automaton that processes symbols from two sequences at the same time, the stimulus (X-sequence) and the response sequence (R-sequence). Under this analogy, M-states process one symbol of each sequence (they match a symbol pair from X and R), X-states process only a stimulus symbol, and R-states process only a response-symbol. The automaton keeps track of two pointers that indicate the current

position in the stimulus sequence as well as the current position in the response sequence. The pointer difference corresponds to the current lag  $dT+e$  (the “automaton” considers all possible lags, weighted probabilistically).

A sequence of hidden states in an MXR-MPH (M, X, and R states) can be depicted as a path in an alignment matrix that spans all possible pairings between stimulus and response (Fig. 2B): An M-State corresponds to diagonal movement along the matrix from position  $(t, u)$ , i.e. position  $t$  in the stimulus sequence and  $u$  in the response sequence, to position  $(t+1, u+1)$ . An X-State corresponds to horizontal movement from position  $(t, u)$  to  $(t+1, u)$  and an R-State corresponds to vertical movement from position  $(t, u)$



to  $(t, u+1)$ . A change in the temporal relationship between stimulus and response (i.e. spike jitter) is reflected in non-diagonal (horizontal or vertical) steps in the alignment matrix (with step size provided by the discretization of stimulus and neural response sequences, Fig. 2B). This 3-state model will be applied to simulated and real data in the next section.

In order to handle state-dependent (switching) neural responses (Fig. 1C), we consider MPHs with several M-states (Fig. 2C). Multiple M states add another dimension to the alignment matrix, which we henceforth call alignment tensor (Fig. 2D). The simplest switching-enabling MPH has two M-states but no X or R states. Each M-state is associated with a particular receptive field to be estimated (here we use ‘receptive field’ in the most general way, independent of linearity and related assumptions). With several M states, spikes can be associated to stimuli via one of several joint probability distributions over spike and stimulus. Probabilistic transitions between M states allow the MPH to switch between receptive fields, which is shown on simulated and real data below.

The general MPH has several M-, X-, and R-States and thus simultaneously permits flexible timing and context dependency. The parameters defining probabilistic transitions into and out of hidden states are:

$A_{ij}$ : Transition probability of transiting from hidden state  $i$  to hidden state  $j$ ,

$I_i$ : Initial probability of hidden state  $i$ ,

$F_i$ : Final probability of hidden state  $i$ .

The parameters defining emission probabilities are:

$b_i(x_t)$ : Emission probability density of the stimulus  $x_t$  given hidden X-state  $i$  ( $x_t$  is a vector and typically spans a window of the stimulus around time  $t$ ),

$b_i(r_u)$ : Discrete emission probability distribution of the response  $r_u$  given hidden R-state  $i$  ( $r_u$  is part of a discrete set, e.g.  $\{0, 1\}$  for spike or no spike),

$b_i(x_t, r_u)$ : Mixed discrete-continuous emission probability of stimulus-response pair  $(x_t, r_u)$  given hidden state  $i$

As emission probability densities associated with X and M states we use multivariate Gaussians or mixtures of Gaussians, respectively. For hidden X-state  $i$  we write the emission probability density as

$$b_i(x) = \sum_{k=1}^K c_{ik} N(x, \mu_{ik}, \Sigma_{ik}),$$

where  $c_{ik}$  is the weight of the  $k^{\text{th}}$  mixture component,  $K$  denotes the total number of mixture components (which may vary for different hidden states, i.e. some  $c_{ik}$  can be zero), and  $\mu_{ik}$  and  $\Sigma_{ik}$  denote the Gaussian mean and covariance matrix of the  $k^{\text{th}}$  mixture component. For M states we keep track of one such multivariate Gaussian for each response state  $r_u$  (each response state is associated with a distinct stimulus emission).

### Special Cases of MPHs

In the following we present detailed analyses of the spike-jitter and response-switching MPH architectures. First, we discuss an MPH with only one M-state (*M-MPH*) and multivariate Gaussian stimulus models. We show that, under appropriate parameter constraints, such a model is equivalent to a 1-dimensional LNP model and describe its relation to linear regression and linear discriminant analysis and the resulting strengths and limitations.

Second, we discuss an extension of that M-MPH to a model that also possesses an X- and an R-state (*MXR-MPH*). We illustrate in simulations how this model can account for spike time jitter and varying latencies on white noise and on natural stimuli. Third, we treat an extension of the M-MPH to multiple M-states (*M<sup>n</sup>-MPH*) and illustrate through simulations how this model can account for switching dynamics and context dependency. All of these models can be cascaded with an additional non-linearity so that they form NNP cascades (as opposed to LNP cascades). In the chapter that then follows, we apply these models to data recorded from single units of cells in the behaving bird.

**The M-MPH with multivariate Gaussian stimulus models and equal covariances.** An MPH characterized by one M-state, multivariate Gaussian stimulus models with shared covariance matrix, and no X and R states is equivalent to an LNP model. In the following we calculate both the linear filter and the LNP non-linearity. Let  $\mu_0, \Sigma_0, \mu_1, \Sigma_1, p_0$ , and  $p_1$  be the parameters of an MPH with one M-state and Gaussian stimulus models, where  $\mu_0, \Sigma_0$  are the mean and the covariance matrix of the stimulus given no spike is emitted and  $\mu_1, \Sigma_1$  are the mean and the covariance matrix of the stimulus given emission of a spike. In this section, we assume identical covariance matrices:  $\Sigma_1 = \Sigma_0 = \Sigma$ .  $p_1, p_0$  are the marginal (or prior) probabilities of a spike and no spike and  $X = x_1, x_2, \dots, x_T$ ,  $x_t \in \mathbb{R}^n$  denotes the stimulus sequence and  $R = r_1, r_2, \dots, r_U$ ,  $r_u \in \{0, \dots, B\}$  the response sequence. In the examples to follow, stimulus and response have the same length,  $T = U$  ( $T \neq U$  can be useful too, for instance, when stimulus and spiking response are differently binned).

In this simple MPH the posterior probability of a spike at time  $t$  is given by

$$\begin{aligned} P(r_t = 1 | x_t = x) &= \frac{P(r_t = 1, x_t = x)}{P(r_t = 1, x_t = x) + P(r_t = 0, x_t = x)} \\ &= \frac{p_1 e^{-0.5(x - \mu_1)^T \Sigma^{-1} (x - \mu_1)}}{p_1 e^{-0.5(x - \mu_1)^T \Sigma^{-1} (x - \mu_1)} + p_0 e^{-0.5(x - \mu_0)^T \Sigma^{-1} (x - \mu_0)}} \end{aligned} \quad (1)$$

By using  $p_0 + p_1 = 1$  and rearranging terms, we can transform Eq. 1 to

$$\frac{1}{P(r_t = 1 | x_t = x)} - 1 = \frac{p_0 e^{-0.5(x - \mu_0)^T \Sigma^{-1} (x - \mu_0)}}{(1 - p_0) e^{-0.5(x - \mu_1)^T \Sigma^{-1} (x - \mu_1)}}. \quad (2)$$

By taking the logarithm on both sides of Eq. 2 we find

$$\begin{aligned} \log\left(\frac{1}{P(r_t = 1 | x_t = x)} - 1\right) &= \log\left(\frac{p_0}{1 - p_0}\right) \\ &\quad - 0.5(\mu_0 + \mu_1)^T \Sigma^{-1} (\mu_0 - \mu_1) + x^T \Sigma^{-1} (\mu_0 - \mu_1). \end{aligned} \quad (3)$$

Hence,  $\log\left(\frac{1}{P(r_t = 1 | x_t = x)} - 1\right)$  is affine-linear in  $x$ . Equivalence between this simple MPH and linear non-linear neural response models follows after applying the sigmoid function,  $\phi(x) = \frac{1}{e^x + 1}$ , on both sides of (3) and assuming  $\mu_0 = 0$ , which yields

$$P(r_i=1|x_i=x)=\phi\left(c-x^T\Sigma^{-1}\mu_1\right), \quad (4)$$

where the constant  $c$  is given by  $c=\log\left(\frac{p_0}{1-p_0}\right)+0.5\mu_1^T\Sigma^{-1}\mu_1$ .

Hence, the posterior spike probability  $P(r_i=1|x_i=x)$  for this MPH agrees with that of an LNP model with linear filter (or receptive field)  $B=\Sigma^{-1}\mu_1$  and non-linearity

$$\phi(c-z). \quad (5)$$

For non-white stimuli, i.e.  $\Sigma \neq I$  with  $I$  the identity matrix, the receptive field  $B$  of the MPH is given by

$$B=\Sigma^{-1}\left(\sum_{t=1}^T r_t \cdot x_t\right), \quad (6)$$

which is known as reverse correlation, corrected spike triggered average [24,25], or simply as linear regression. Moreover, for white stimuli, i.e.  $\Sigma=I$ , the receptive field  $B=\mu_1$  of the MPH is the spike triggered average (STA) - the mean of the spike triggered ensemble. Hence, the M-MPH performs a linear regression cascaded with a sigmoid non-linearity and is equivalent to a one dimensional LNP model with a sigmoid non-linearity (Eq. 5).

Note that the M-MPH's receptive field (Eq. 6) always corresponds to the linear regression solution. Consequently, the M-MPH's receptive field estimate is optimal whenever linear regression is the correct model. In particular, it follows that the MPH parameter estimates can be optimal even when the spike triggered ensemble and its complement are non-Gaussian - for instance in case of white noise stimuli and a threshold non-linearity [1,26]. The same is true when the overall stimulus distribution is non Gaussian - for instance for non-Gaussian natural stimuli and a linear "non-linearity" [26]. However, although the receptive field estimate does not depend on Gaussian assumptions, the non-linearity (Eq. 5) does, see e.g. [26]. In [26], the authors suggest to re-estimate the non-linearity (decision boundary) of a linear discriminant model to obtain a non-linearity estimate not corrupted by Gaussian assumptions.

Inspired by [26], we cascade the MPH with an additional non-linearity that can be estimated from the data subsequent to the estimation of the MPH (compare Materials and Methods for details of the estimation). Such cascading is standard practice for neural response models [1,27,28] and is also part of LNP models. The cascaded M-MPH with shared covariance can fit any one-dimensional LNP model on white-noise data (provided that the operation of the LNP model leads to a change of mean of the spike triggered ensemble, which is the case for all monotone non-linearities and for most others) [1]. However, for non-white data and certain nonlinearities, the linear regression estimate can be biased [1,2], so that MPHs too will infer an incorrect receptive field (see also below, where we apply MPHs to natural stimuli). In these cases, using Gaussian mixtures as stimulus models might be advisable (see discussion).

**M-MPH with free covariance matrices and Gaussian mixture models.** One interesting extension of that simple MPH results from assuming  $\Sigma_1 \neq \Sigma_0$ . This case is analogous to quadratic discriminant analysis [26]. The MPH implements a model quadratic in  $x$ . One further extension is to use mixtures of Gaussians as emission distributions instead of individual Gaussians (see discussion), in which case  $g$  is generally not quadratic anymore.

## The MXR-MPH and Its Application to Simulated Data with Spike-Time-Jitter

In the following we demonstrate the ability of MXR-MPHs (Fig. 2A) to recover the correct receptive field (RF) on simulated data with noisy latencies, i.e. spike-time-jitter. When predicting spiking probabilities on novel data, the MXR-MPH outperforms purely spike-triggered methods.

For the MXR-MPH, we denote the stimulus means and covariance matrices in the M-State by  $\mu_{M0}, \Sigma_{M0}$  (non-spiking) and  $\mu_{M1}, \Sigma_{M1}$  (spiking) and in the in the X-State by  $\mu_X, \Sigma_X$ . In accordance with the section on LNP equivalence and to simplify this general MPH, we introduce the following parameter constraints: First, we fix all covariance matrices to identity matrices:

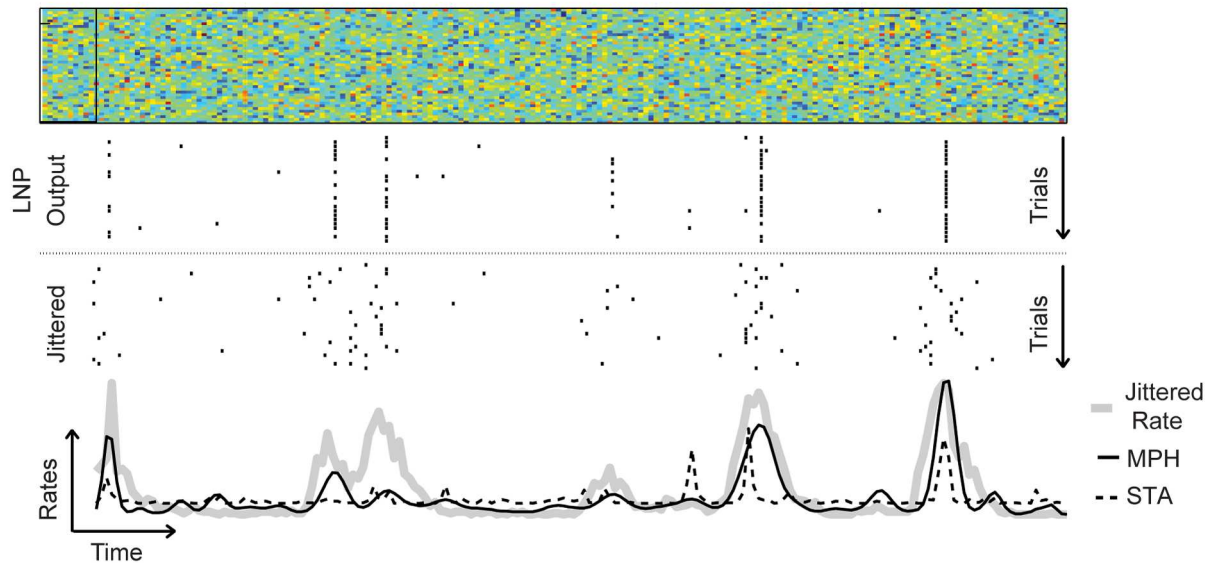
$$\Sigma_{M0} = \Sigma_{M1} = \Sigma_X = I.$$

Second, we fix the means  $\mu_{M0}$  and  $\mu_X$  to zero (equal to the mean of the stimulus ensemble). Third, we do not allow the R-state to generate spikes (zero spike emission probability) because we require that each spike is matched to a stimulus (note that if we allowed the R-state to generate spikes, the model would distinguish between spikes generated by the M-state and spikes generated by the R-state - such distinction could be used for distinguishing stimulus driven from spontaneous activity, which was not our focus). Given these parameter constraints, the remaining free parameters in the model are the receptive field  $\mu_{M1}$  (the mean of the spike triggered stimulus ensemble, Fig. 2C), and the transition probabilities among X, R, and M-state. After training the MPH, the jitter statistics are implicit in the model's parameters; below we show how to explicitly compute the jitter statistics for natural stimuli.

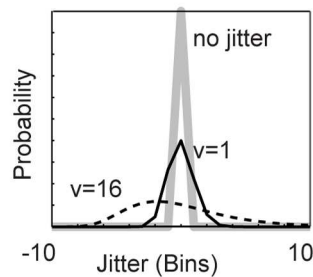
**Application to white data.** To test the model, we first created artificial data by sampling spike trains from an LNP model in response to a white noise stimulus consisting of  $10^5$  time bins (arbitrary timescale), and 42 dimensions or stimulus channels (e.g. pixels on a one dimensional screen or frequency bands in a spectrogram). We split the stimulus into 500 sequences of equal duration and generated 25 trials of LNP spiking responses for each of those sequences (Fig. 3A). The LNP model was composed of one linear filter that spanned 11 time bins (Fig. 3C) and a sigmoid nonlinearity (red line in Fig. 3G). We generated a total of 28702 LNP spikes ( $\sim 0.015$  spikes/bin). We randomly jittered the individual LNP spikes (Fig. 3A) with i.i.d. spike shifts drawn from a discretized log-normal distribution with zero mean (Fig. 3B). The variance of that distribution controls the total amount of jitter: As variance increases, the distribution becomes more asymmetric with a heavy right-side tail (we choose such an asymmetric jitter distribution to increase the difficulty of the problem). To 450 of the 500 stimulus-response sequences we fitted both an MXR-MPH and a reverse correlation model (resulting filters are depicted in Fig. 3C). The remaining 50 sequences served as validation set.

The STA, which is the optimal solution in case of jitter-free data, yielded a poor approximation of the true receptive field (Fig. 3C). The main problem for STA was that the (jittered) spike-triggered stimulus ensemble did not separate well from its complement (Fig. 3D, middle) when projected onto the underlying (true) RF. In contrast, the receptive field  $\mu_{M1}$  estimated using the MXR-MPH (Fig. 3C) was very close to the true RF, and the reconstructed spike-triggered ensemble (computed by aligning stimulus and response through the generalized Viterbi algorithm, see Materials and Methods) was well separated from the full

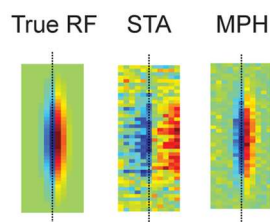
## A Stimulus and Spiking Responses



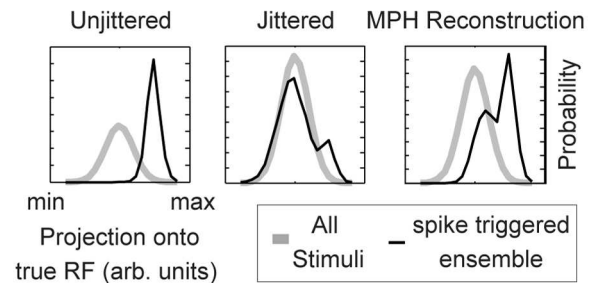
## B Jitter Kernels



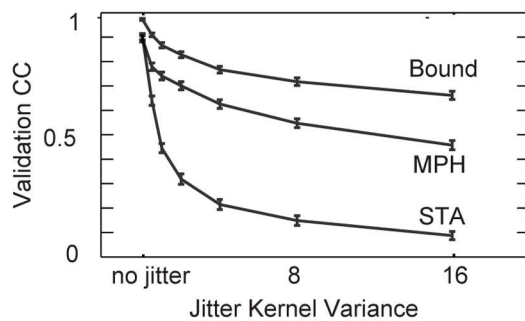
## C Receptive Fields Estimated on Jittered Data



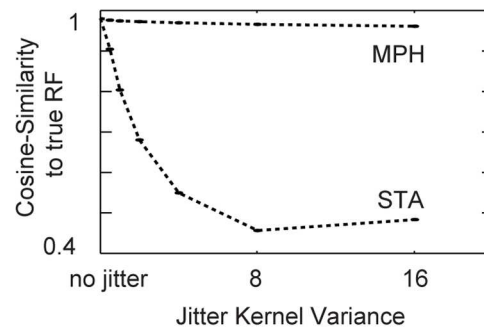
## D Spike Triggered Ensembles



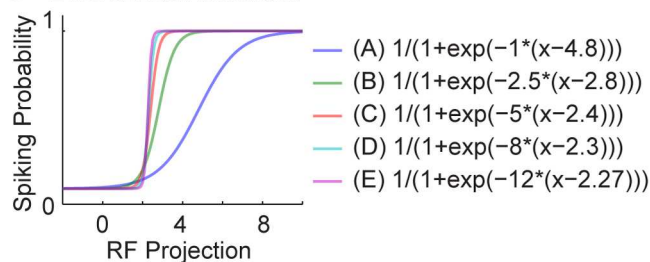
## E Prediction Quality



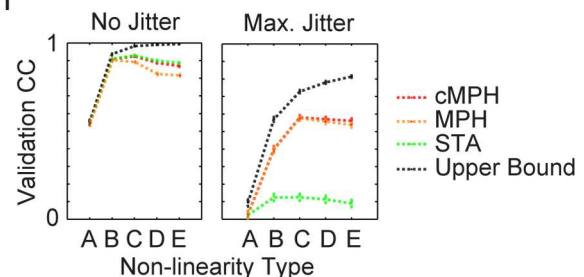
## F RF-Reconstruction Quality



## G Different Non-linearities



## H



**Figure 3. The MXR-MPH applied to white noise stimuli and spike-time-jitter.** (A) A white noise stimulus (top) with spiking responses (black bars) generated by an LNP-type model neuron (LNP output, the LNP RF size is indicated by the black rectangle). The jittered versions (jittered) of the LNP spike trains with corresponding firing rate (thick gray line) are shown below. The MPH estimate of firing rate (black full line) is more accurate than



the STA estimate (dashed line). (B) Applied spike jitter is i.i.d. among spikes and log-normally distributed with zero mean (3 different jitter distributions are shown; they differ in terms of variance  $v$  and symmetric/asymmetric shape). Results for the jitter kernel with variance  $v=16$  are shown in panels A, C and D. (C) RFs estimated through STA on unjittered spikes (true RF), STA on jittered spikes (STA), and MPH on jittered spikes (MPH). The STA RF is blurred whereas the MPH RF closely resembles the true RF. Dotted black lines indicate the midpoints of the RFs. (D) Projections of all stimuli (gray lines) and the spike triggered stimulus ensembles (black lines) onto the underlying (true) RF for the unjittered spikes (left), the jittered spikes (middle), and the MPH reconstruction (right), obtained via dynamic alignment using the generalized Viterbi algorithm). (E) *Response prediction*. To evaluate the models we computed correlation coefficients (CCs) between predicted and actual firing rates on the validation set and for different jitter variances. For small spike jitter, performances of STA and MPH are comparable. As the jitter magnitude increases, STA performance drops much more severely than does MPH performance. Also shown is an upper bound for the CC computed by sampling and cross-correlating jittered responses. (F) MPH robustness to jitter is demonstrated also when assessed as similarity between the estimated RF and the true RF (similarity computed as normalized scalar product, i.e. cosine of angle between RFs). (G) We assessed the influence of different non-linearities (labeled A–E, ordered by steepness) on prediction quality for both the MPH as well as the cascaded MPH (cMPH). (H) Shallow non-linearities decrease the upper bound of prediction quality (black line) as well as the MPH (red lines) and STA (green line) performance for the unjittered (left) and the jittered case (right). The cascaded MPH (red line) shows slight improvements over the non-cascaded one (dotted red line). doi:10.1371/journal.pcbi.1003508.g003

stimulus ensemble (Fig. 3D, right). We ran the spiking-probability inference algorithm outlined in Materials and Methods on the independent validation set. The MPH-predicted spike responses were in general much better than STA-predicted responses (Fig. 3E, instead of using correlation coefficients, we could have evaluated performance through the average likelihood of the models given the data; we opted for CCs to ensure easy interpretability and connect to existing literature). For small jitter, MPH and STA responses were equally good; however, with increasing jitter, the MXR-MPH performance dropped much less than that of STA. Similar superiority of the MXR-MPH was also seen in RF estimation, evaluated in terms of the angle between estimated and true RFs (Fig. 3F, to discount for arbitrary shifts in RF position that could be induced by the asymmetric jitter kernels we also designed a shift-invariant measure by time-shifting the estimated RF relative to the true RF and considering only the minimal angles; this gave virtually identical results).

We elucidate the influence of various non-linearities, by having evaluated MPHs and STA-models for various sigmoidal non-linearities of the (true) LNP model (Fig. 3G, the non-linearities were chosen such that the resulting models each yield an average rate of about 0.015 spikes/bin). We found that RF estimation and response prediction of the MPH was the better the steeper the non-linearities (Fig. 3H). Furthermore, the difference between cascaded and non-cascaded MPH is not large (Fig. 3H). The improved performance for steep nonlinearities is to be expected because the non-linearity is needed to separate the action of the linear filter from the action of the jitter kernel in the underlying LNP model (i.e. without non-linearity the linear kernel of the LNP model and the jitter kernel simply act as two subsequent linear operations which no longer can be uniquely disentangled).

**Application to natural stimuli.** We also tested the MXR-MPH on jittered responses to natural stimuli (a problem that, to our knowledge, has not yet been addressed in the literature). We sampled spike trains from an LNP model in response to spectrograms of birdsongs (Fig. 4A). We used 250 zebra finch songs (50 of the songs served as a validation set), yielding a total of 66025 time bins (4 ms each) and 20332 generated spikes (mean rate/bin  $\sim 0.012$ ).

We fixed the model covariance matrices to the covariance  $\Sigma$  of the stimulus ensemble:

$$\Sigma_{M0} = \Sigma_{M1} = \Sigma_X = \Sigma \neq I.$$

Furthermore, we fixed the means  $\mu_{M0}$  and  $\mu_X$  to the actual stimulus mean. As for white noise stimuli, the MPH performed better than reverse correlation on RF estimation and response prediction (Fig. 4E, 4F).

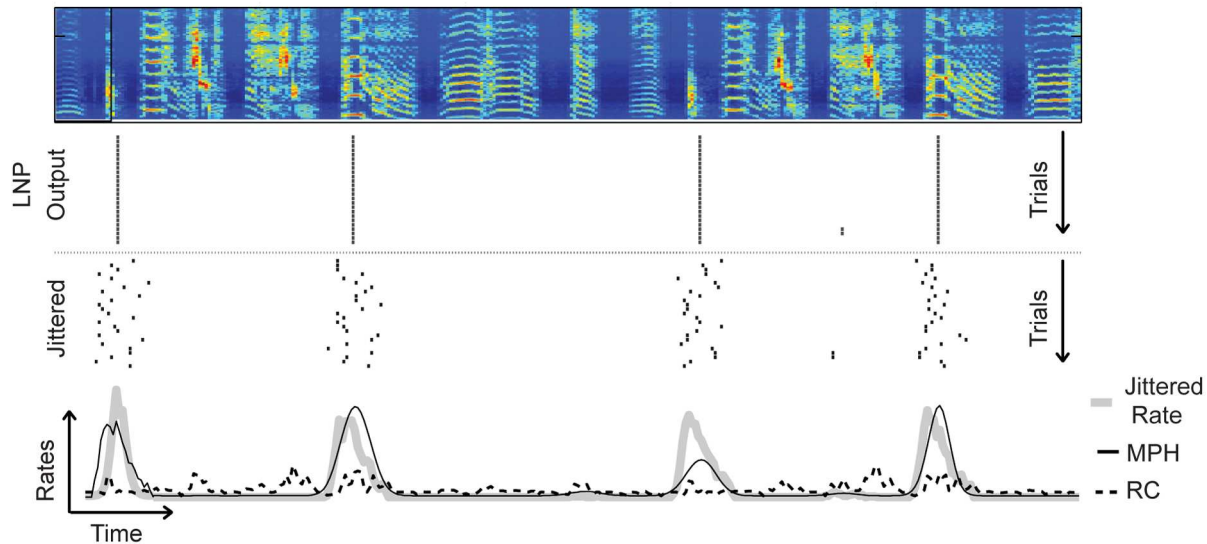
We computed the jitter statistics via the alignment kernel of the MXR-MPH (the alignment kernel is a weighted average of spike shift counts associated with each possible hidden state sequence (i.e. each path in the alignment tensor, Fig. 2B) where the weights correspond to the respective probabilities of the hidden state sequences given model and data, see Materials and Methods). Our simulations showed that the model did not over-fit the data by detecting jitter when none was present (Fig. 4B, left) and that the alignment kernel could be estimated quite well even when jitter was large (Fig. 4B, right).

The MPH allows for stimulus-response modeling both for correlated and uncorrelated jitter: Correlated jitter can be accounted for by decreasing the transition probabilities onto X and R-states, which in turn decreases the probability of non-diagonal movement in the alignment tensor (thus leading to correlated stimulus-response lags across successive spikes). To model uncorrelated jitter, the transition probabilities can be chosen such that the likelihood of a chunk of the stimulus-response pair being modeled with only X- and R-states equals the likelihood of modeling it with M-states only. In that case, constant time lags and changing time lags between successive spikes are equally likely and jitter is uncorrelated (provided that successive spikes are further apart than the jitter size).

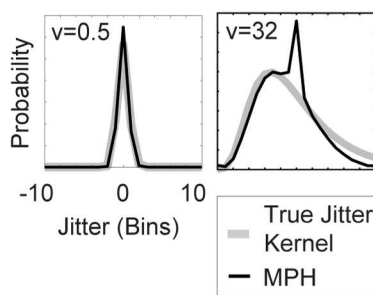
## The M<sup>2</sup>-MPH and Its Application to Simulated Data with Switching Dynamics

MPHs with several M states support *context dependency*. They can model multiple stimulus-response relationships associated, for instance, with distinct behavioral states of an animal. To demonstrate this flexibility of MPHs, we simulated a neuron that randomly switches (according to a Markov process with equal probabilities) between two linear-nonlinear models (each defined as in the previous section), i.e., neural responses were governed by a hidden state sequence that determined which receptive field was active at any given time (Fig. 5A). We generated responses of this artificial neuron to 100 white noise sequences, each spanning 1000 time bins (arbitrary timescale), and 21 dimensions or stimulus channels. For each sequence, spike responses were generated using the switching LNP model, resulting in a total of 4374 spikes on average (mean 0.044 spikes/bin). We generated and tested data for different RF combinations, characterized by different rotations in the plane of one of the RFs (Fig. 5B, left). To uncover the hidden switching dynamics and the two RFs, we trained a Gaussian M<sup>2</sup>-MPH with two M-states A and B (Fig. 2C). Gaussian parameters were constrained in the following way: First, we imposed zero means:  $\mu_{A0} = \mu_{B0} = 0$ ; and second, we fixed all covariance matrices to the identity matrix. The remaining free parameters of the model were the means  $\mu_{A1}$  and  $\mu_{B1}$  of the spike-triggered ensembles (i.e.,

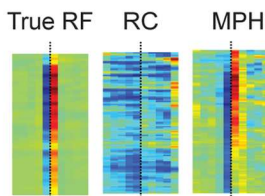
## A Stimulus and Spiking Responses



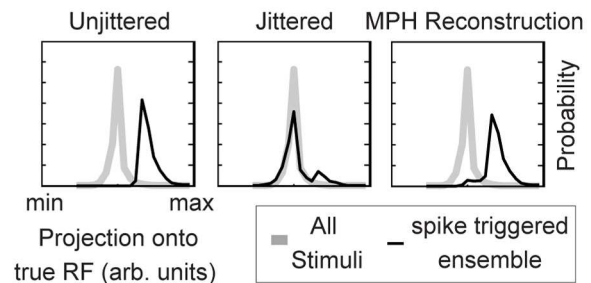
## B Reconstructed Jitter Kernels



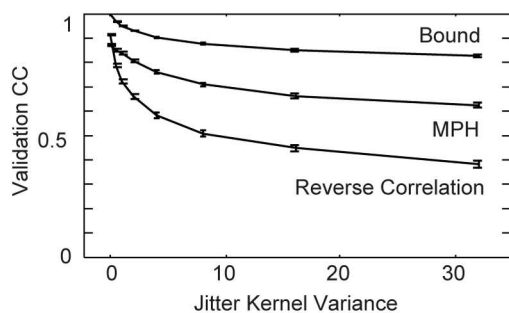
## C Receptive Fields Estimated on Jittered Data



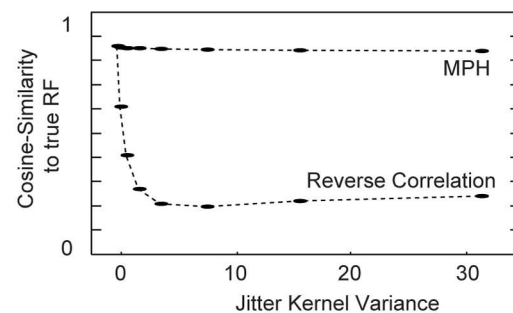
## D Spike Triggered Ensembles



## E Prediction Quality



## F RF-Reconstruction Quality



**Figure 4. The MPH applied to natural stimuli and jittered spike responses.** (A) An example log-spectrogram of zebra finch song (top, high sound amplitudes in red and low amplitudes in blue), spiking responses generated by an LNP-type model (middle, LNP output), their jittered versions (below), and the corresponding jittered firing rate (bottom, gray line). The MPH-predicted response (MPH, full line) of the jittered firing rate is more accurate than the reverse correlation prediction (RC, dashed line). (B) Applied spike jitter is i.i.d. among spikes and log-normally distributed with zero mean. Two different jitter distributions are shown, they differ in terms of variance  $v$  and symmetric/asymmetric shape (gray curves  $v=0.5$  left, and  $v=32$  right). The MPH-estimated jitter kernels are shown in black. The MPH misses some jittered spikes (right), as revealed by the excessive peak at zero time lag. Results for the jitter kernel with variance  $v=32$  are shown in panels A, C, and D. (C) RFs estimated through reverse correlation for unjittered data (true RF), jittered data (RC) as well as the MPH receptive field estimate (MPH). The STA RF is blurred whereas the MPH RF closely resembles the true RF. Dotted black lines indicate the midpoints of the RFs. (D) Projections of all stimuli (gray lines) and the spike triggered stimulus ensembles (black lines) onto the underlying (true) RF for the unjittered spikes (left), the jittered spikes (middle), as well as the MPH reconstruction (right, obtained via dynamic alignment using the generalized Viterbi algorithm). (E) Correlation coefficients (CCs) between predicted and true firing rates on the validation set for different jitter variances. Also shown is an upper bound for the CC computed by sampling and cross-correlating jittered

responses. For small overall jitter, performances of reverse correlation and MPH are comparable. As the overall jitter magnitude increases, reverse correlation performance drops much more severely than does MPH performance. (F) RC performance drops even stronger when assessed in terms of similarity between the estimated and the true RFs. doi:10.1371/journal.pcbi.1003508.g004

the receptive fields) and the transition probabilities between the two states (which reflect the switching statistics).

We compared the MPH with STA and STC [1,6] models. For all three models, we computed RF estimates and the response prediction performance. The trained MPH faithfully recovered both RFs, whereas the (single) RF estimated of STA consisted of a superposition of the two RFs, and the RFs estimated using STC were severely corrupted by noise in most cases (Fig. 5B). As a result, the MPH predicted responses better on an independent validation set than did STA and STC (Fig. 5C, averaged over 10 runs). We assessed the quality of the recovered RFs of all three methods and all 37 tested rotations by matching each original RF to the recovered RF with smallest distance and by averaging the two distances. The MPH recovers the RF much better than STA or STC (Fig. 5D, average over 10 runs; drops in MPH-reconstruction quality are due to local minima, compare figure text).

The degraded performance of the STC model has two reasons. First, the covariance of the spike triggered ensemble needs to be reliably estimated (with quadratically many degrees of freedom as there are stimulus dimensions compared to a linear number of degrees of freedom for the M<sup>2</sup>-MPH). Second, the linear filters uncovered by STC are orthogonal [1], whereas the M<sup>2</sup>-MPH is not constrained in this way.

It is possible to show that the M<sup>2</sup>-MPH firing rate  $p_t$  to a stimulus is given by

$$p_t = \lambda_A \phi_A(X_t RF_A) + \lambda_B \phi_B(X_t RF_B),$$

where  $\phi_1, \phi_2 : \mathbb{R} \rightarrow [0,1]$  are two non-linearities and  $\lambda_A$  and  $\lambda_B$  are the prior probabilities of hidden states  $A$  and  $B$ , respectively. The STA model, on the other hand, is bound to model firing rates as

$$p_t = \phi(X_t(\lambda_A RF_A + \lambda_B RF_B)).$$

An extreme example that illustrates the failure of RF estimation with STAs is a neural response model that pools over two filters  $RF_A = -RF_B$  and  $\lambda_A = \lambda_B = 0.5$ . In that case the estimated RF using STA is uniform and has no predictive power at all, unlike the MPH (e.g. Fig. 5C, rotation angle 180°). A less extreme but potentially more relevant case is that of complex cells in primary visual cortex with overlapping excitatory and inhibitory oriented receptive subfields (such cells are often modeled by pooling over four oriented filters that are phase shifted 0°, 90°, 180°, and 270°, respectively [29–31]). A switching M<sup>4</sup>-MPH with four M-states can recover these phase shifted filters, whereas STA yields only a blurred RF.

### Application to Songbird Spike Data

To demonstrate that MPHs work well in practice even when the amount of available data is small and the true spike generating process is unknown, we apply the MXR- and the M<sup>n</sup>-MPH to extracellular spike data recorded in the forebrain nucleus interface of the nidopallium (Nif) of songbirds (Fig. 6A). Nif is a higher-order song-control nucleus; lesion and inactivation studies have shown that Nif exhibits both sensory and motor functions [32–35]. Multi-unit Nif activity is generally strongest shortly before and

during syllable production and weakest during the times corresponding to silent intervals between syllables [36,37]. These findings suggest a pre-vocal role of Nif spikes during song, prompting us to expect in singing birds a negative latency of Nif spikes relative to song (spikes precede sounds, as opposed to a positive latency that would result if Nif firing was sensory during vocal production). Due to the difficulty of recording in singing animals, available spike trains are relatively short (the average total spike train duration was 73 s per cell) and contain few spikes (~1500 spikes per cell).

To investigate latencies of Nif single-unit spikes relative to song, we first fitted an LNP model using reverse correlation (RC, Fig. 6C, left). To overcome problems of over-fitting (due to the limited amount of data available) we used a regularized version of the stimulus covariance matrix:

$$\Sigma' = \Sigma + \alpha I, \quad \alpha = \frac{\text{Tr}(\Sigma)}{n}, \quad (7)$$

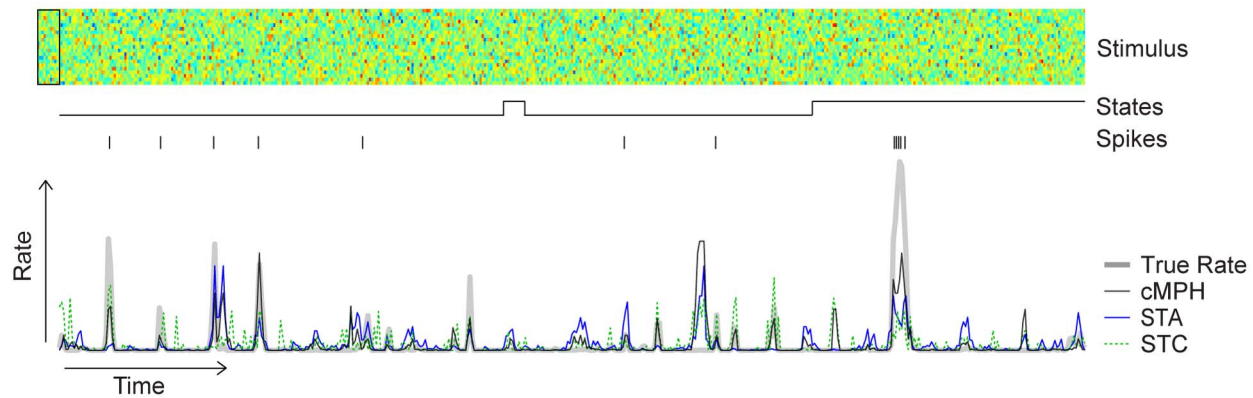
where  $n$  denotes the number of stimulus dimensions,  $\Sigma$  denotes the unregularized stimulus covariance matrix and  $\alpha$  denotes its normalized trace (such regularization yielded better generalization performance). Next, we trained a MXR-MPH on large 0.25 s song spectrogram windows (with covariance matrices in M and X states fixed to the regularized stimulus covariance matrix  $\Sigma'$  in Eq. 7). The MPH RF was similar to the reverse correlation RF (Fig. 6B), but it reflected more clearly that the cell fired before sounds and not thereafter (consider for example the stronger inhibitory band near 10 ms). MPH and reverse correlation encoding performances on a test set were comparable (Fig. 6D, left data points). Note that by construction, differences between MXR-MPH and reverse correlation RFs arise from spike-time-jitter.

To characterize response latencies (and jitter) we estimated probability distributions of the temporal offset  $\tau$  between stimulus and response in M-states via the *alignment kernel* (Fig. 6B, left). Negative lags in the alignment kernel imply that spikes occur before corresponding events in the stimulus, whereas positive lags imply that spikes occur thereafter. The alignment kernel was centered at a small negative time lag and exhibited a small temporal spread, revealing high temporal precision of Nif spike trains. Predicted responses (5-fold cross validation) for the reverse correlation model and the MXR-MPH were equally good (Fig. 6D, left data points), confirming high temporal precision of Nif spike trains.

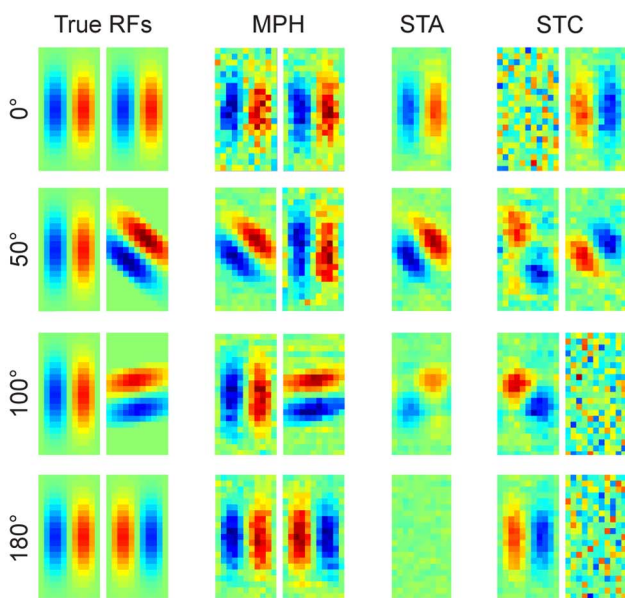
The MPH allowed us to strongly reduce model complexity by shrinking linear filters (RF sizes) down to less than 30 ms. For such short RFs, the cell latency is reflected entirely in the alignment kernel. Based on the RF estimate in Fig. 6B and 6C, we expected the jitter kernel to be centered near -30 to -40 ms. Indeed, the kernel peaked near -40 ms (Fig. 6C, right), implying that the MPH aligned spikes to portions of the stimulus occurring about 40 ms after the spike, suggesting a premotor function of this cell and thus agreeing with the hypothesized premotor function of Nif.

Additionally we trained an M<sup>n</sup>-MPH with various numbers of states on the same Nif cell (unlike for the M<sup>2</sup>-MPH applied in the section on switching dynamics we did not constrain the means). The M<sup>n</sup>-MPH showed modest improvements over reverse correlation, its peak validation CC occurred at 8 states (Fig. 6D), suggesting that this Nif cell fires prior to several distinct song features.

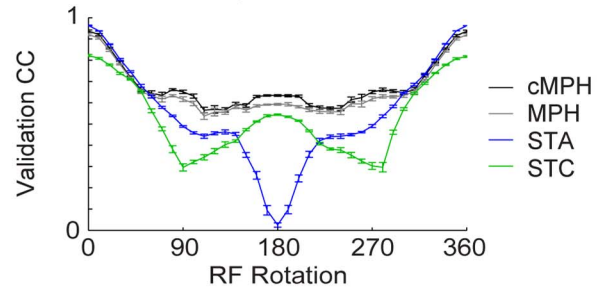
## A Stimulus and Spiking Responses



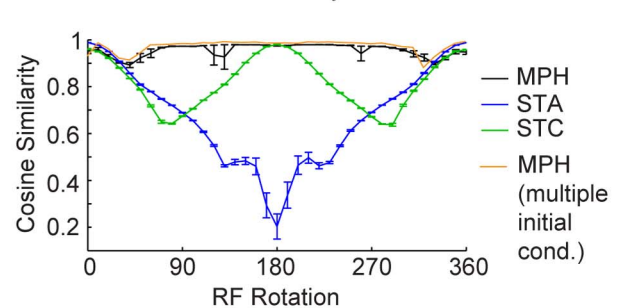
## B Estimated Receptive Fields



## C Prediction Quality



## D RF-Reconstruction Quality



**Figure 5. The MPH applied to white noise stimuli and switched responses.** (A) A white noise stimulus (top), the randomly switched states of a switching LNP model (middle, black curve), and the observed spike train (middle, black rasters) and firing rate (bottom, gray line). The MPH-predicted firing rate (bottom, black line) to a test stimulus is closer to the observed firing rate than is the STA prediction (blue line) or the STC prediction (dotted green line). (B) The MPH RF estimates (MPH, 2<sup>nd</sup> column) capture well the underlying true RFs (True RFs, 1<sup>st</sup> column) for all relative angles, unlike the STA RF estimates (STA, 3<sup>rd</sup> column) or the STC RF estimates (STC, 4<sup>th</sup> column). (C) We evaluated the models by computing CCs between predicted and observed firing rates on a validation set and for different pairs of LNP filters that were generated by rotating one of the RFs. The cascaded MPH (black line) performs slightly better than the non-cascaded MPH (gray line). Both MPHs perform better than STC (green line) and STA (blue line). (D) Quality of RF reconstruction, shown is the cosine angle between true and model RFs (compare main text). The MPH reconstructed the true RFs more faithfully (black line) than did STA (blue line) and STC (green line). The occasional drops in MPH performance (larger error bars) are due to local optima that can be circumvented by starting the MPH-parameter optimization from different initial conditions (the orange line is from the best model – in terms of likelihood on the training set – out of 3 initial conditions). Both, panels (C) and (D) show average results from 10 simulations (with standard errors indicated). doi:10.1371/journal.pcbi.1003508.g005

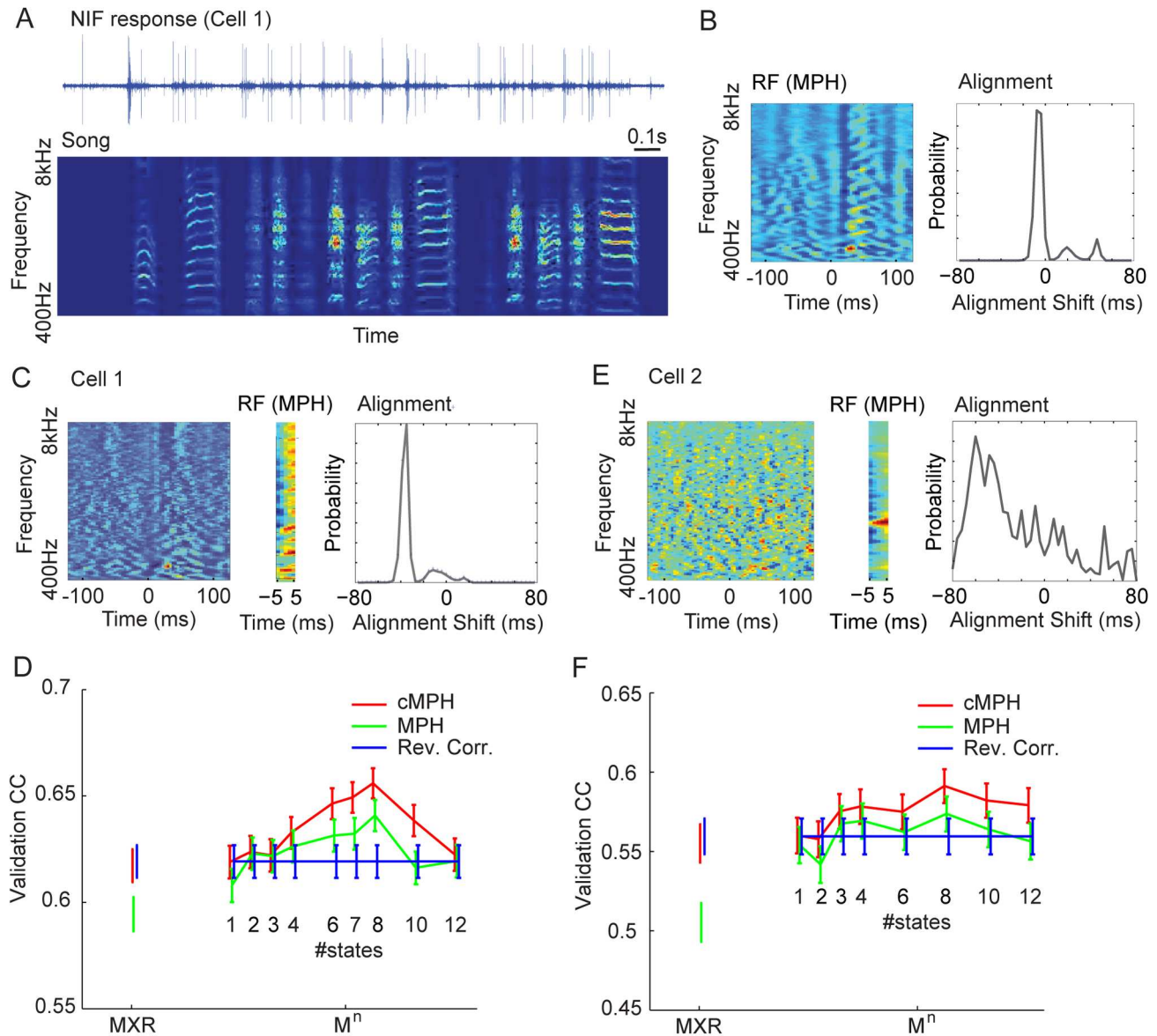
We also analyzed data for another recording site in NIF, composed of 54 s of singing with concurrent spiking (1659 spikes, about 60 stereotyped song motifs). The RF estimated using reverse correlation (Fig 6E, left) revealed diffuse spectrotemporal tuning, making it difficult to decide whether this cell is sensory or motor in function. By contrast, the MPH alignment kernel (Fig. 6E, right) quite clearly revealed a motor function in this cell, evidenced by the predominance of negative alignment shifts. The MPH RF showed a rather narrow frequency tuning near 2.6 kHz (Fig. 6E, middle). Encoding performance for the MXR-MPH with large RF was again

similar to reverse correlation (Fig. 6H, left data points), yet an M<sup>P</sup>-MPH yielded slightly superior performance (Fig. 6H, right data points).

## Discussion

We introduced a novel technique for neural response modeling and receptive field estimation that overcomes limitations of fixed stimulus-response relationships. We proposed to view neural coding as an alignment problem that can be tackled by mixed pair hidden Markov models (MPHs), which jointly model





**Figure 6. The MXR- and  $M^n$ -MPH applied to single-unit activity in NIF of a singing zebra finch.** (A) Raw extracellular voltage trace time-aligned to a log-power sound spectrogram of a zebra finch song (high sound amplitudes in red and low amplitudes in blue). (B) The MXR-MPH's RF estimate (left, high and low sound amplitudes in red and blue respectively). The red blob at about +30 ms is an indication that this cell is premotor. The width of the window is  $\sim 0.25$  s. The MXR-MPH's alignment kernel (right) is concentrated near  $-10$  ms, yielding a total lead of Nif spikes on song of about 40 ms. (C) The RF estimated with reverse correlation (left) is similar to the MXR-MPH's RF. Middle: RF and jitter kernel of an MXR-MPH with much narrower RF window (about 10 ms wide). The total dimension of the RF is 605 (5 columns times 121 rows). Because the RF is so narrow, the spike latency is now clearly reflected in the alignment kernel (right), centered around a negative alignment shift of about 40 ms, implying that the model aligns spikes to portions of the song that occur about 40 ms after the spike. Hence, the alignment kernel strongly suggests a premotor function of this cell. (D) Predictions (5-fold cross validation) of the MXR-MPH (left, red bar) are similar to reverse correlation (blue bar). Using the non cascaded version (green bar) yields a slight drop in performance. An  $M^n$ -MPH yields a modest improvement in prediction performance (right, peaking at 8 states) in both the cascaded (cMPH) and non-cascaded forms (MPH, error bars depict 95% confidence intervals). (E) Results for a different data set (a different cell producing 1659 spikes during about 54 s of song data containing about 60 song motifs). The RF estimated using RC reveals diffuse spectrotemporal tuning, making it nearly impossible to decide whether this cell is sensory or motor in function. By contrast, the MPH alignment kernel (right) quite clearly reveals a motor function in this cell, evidenced by the predominance of negative alignment shifts. Also, the MPH RF shows a rather narrow frequency tuning near 2.6 kHz (middle). (F) The MXR-MPH firing-rate predictions for this cell were comparable to reverse correlation predictions;  $M^n$ -MPHs again yield a modest improvement in prediction performance. doi:10.1371/journal.pcbi.1003508.g006

stimulus and response and can naturally account for noisy or systematically varying latencies as well as for context dependent neural codes that depend on internal (hidden) states. Discrete pair HMMs have been used in the context of gene alignment to find corresponding parts in related gene sequences [38]. To

our knowledge they have not yet been applied to response modeling.

We demonstrated that simple MPHs with Gaussian stimulus models and a fixed shared covariance matrix are equivalent to one dimensional LNP models with sigmoid non-linearity and we



extended these basic MPHs to allow flexible timing and context dependency. Thereby MPHs endow standard RF estimation techniques such as spike triggered averaging (STA) and reverse correlation with *flexible timing* and *context dependency*. We tested our approach on simulated and real data and demonstrated the benefits of alignment in terms of improved predictability of simulated and real neural responses, improved receptive field estimates as well as the capability of estimating jitter latency statistics and switching states.

Key properties of MPHs are: 1) X- and R-states that model stimulus or response alone and allow for *flexible timing* via dynamic temporal alignment, and 2) M-states that allow for *context dependency* via model switching. Using our estimation techniques, these three types of states can be freely combined in a highly flexible approach to neural coding and decoding without the need to develop additional algorithms.

We derived MPH parameters estimation updates for Gaussian mixture models with unrestricted covariance matrices (Materials and Methods). The (non-mixture) Gaussian MPHs we studied performed well in simulations (including natural stimuli), even though the assumption of Gaussian stimulus models can be violated by natural stimuli [24,39]. In these cases, mixtures of Gaussians can be useful to approximate arbitrary stimulus distributions and overcome problems of receptive field biases [27]. Beyond mixtures of Gaussians, EM update equations for other mixture families are known as well [40] and could be adapted to MPHs.

Other modeling approaches have been pursued to estimate neural responses in the presence of spike time jitter [15,16,41]. One approach is to simultaneously estimate the jitter distribution and the RF using the EM algorithm [16]. This technique has been successfully applied to white noise stimuli (identity covariance) [16], but not to stimuli with non-identity covariance, i.e. natural stimuli. Furthermore, in [16] the jitter of adjacent spikes is assumed to be independent – an assumption that might be violated in cases where jitter depends on slowly varying internal states or is correlated for other reasons. The dynamic alignment technique we present here generalizes these approaches in two ways. First, in MPHs there is no need to constrain the stimulus covariance matrix, so that natural stimuli can be readily processed. Second, MPHs can account for correlated as well as uncorrelated jitter among adjacent or nearby spikes and thus allow modeling of both systematically and slowly varying spike latencies. Furthermore, in [16] the jitter distribution is explicitly assumed to be of Gaussian form whereas the jitter distribution of the MPH is implicit in the transition probabilities and has degrees of freedom commensurate with the number of hidden states and their transitions.

The ability of MPHs to emulate switching models is particularly useful given that switching dynamics are important in many neural systems. A number of other approaches have been introduced to handle response switching and context dependency. Several of them are based on hidden Markov models [42–48]. The hidden states in these models typically reflect neural activity but not the stimulus. Models with hidden states that reflect both stimulus and response, such as switching Kalman filters [49] or generalized linear models with hidden states [50,51], have also been proposed. These models are similar to MPHs with only M-states but no X- and R-states. Furthermore, our approach extends these models in that stimulus-response relationships within each hidden state can be quadratic (single Gaussians, unconstrained covariance matrices) or formed by Gaussian mixtures. Another way of modeling context dependencies are “multi-linear” models encompassing a multiplicative context term (by itself modeled through a “multi-linear” model) that depends on the projection of the stimulus (in some

time window) onto a set of basis functions [52]. MPHs complement such approaches by allowing more complex types of contextual influence via the underlying Markov structure. This is also an advantage over techniques like spike triggered covariance that can recover multiple filters [1,6,53–55] but cannot attribute Markovian dynamics to the individual filters. For instance, MPHs allow for context effects over very long time scales, context effects depending on hidden neural states such as up and down states (in this case MPHs also allow to infer the up and down states, for instance through the generalized Viterbi algorithm), and left-to-right HMMs [40] can incorporate behavioral context in stereotyped motor actions such as birdsong.

MPHs can bridge between data analysis and theories of neural function. Some theories of cortical function assume discrete modules of computation and representation [56,57], for example synfire chains [47,58] or, more generally, cell assemblies. In these theories, the role of neural activity does not only depend on the identity of the neuron but also on the (hidden) identity of modules the neuron belongs to at a certain time.

The MPHs we developed to align stimulus and neural response are based on stimuli represented with continuous probability densities and neural activity represented with discrete probabilities. It is noteworthy that both fully continuous pair HMMs that align two continuous sequences and fully discrete pair HMMs also have interesting applications. For instance, we have shown previously that a fully continuous pair HMM can be used to align the songs of a juvenile bird to the song of the bird’s tutor in order to identify the parts of the song that were copied and the locations where insertions were made [59]. We have also demonstrated how fully discrete pair HMMs can be used to align spike trains [59]: by learning a discrete pair HMM on pairs of related spike trains, we obtained a “distance” measure between spike-trains, thereby generalizing state of the art spike train metrics [60].

MPHs are useful for both neural encoding and decoding. We presented algorithms for inferring neural responses and their probabilities given the stimulus (encoding). However, by symmetry of MPHs, the inference algorithms we designed can in principle be “inverted” to estimate the stimulus given neural activity (decoding) so that decoding and encoding of brain activity essentially have become the same problem.

MPHs are based on classical hidden Markov models and learning and inference algorithms other than the EM algorithm are readily available. For instance, for model parameter estimation we could have used (much faster) Viterbi training [38] or we could have optimized criteria other than data likelihood [61]. Also, there exists a large variety of very powerful analytical and computational tools developed for classical hidden Markov models that can be adapted to MPHs [61–63].

We will make a code package for fitting MPHs available through our website ([www.ini.ch/~skollmor](http://www.ini.ch/~skollmor)).

## Materials and Methods

### Ethics Statement

All experiments were carried out in accordance with protocols approved by the Veterinary Office of the Canton of Zurich, Switzerland.

### Short Introduction to Hidden Markov Models

We provide a short introduction to “normal” hidden Markov models and the associated terminology for readers unfamiliar with them.

Consider two dies at a game of chance, one with equal probabilities for its six faces (fair die) and the other with unequal probabilities (loaded die). Suppose that their holders can exchange dies for one another without you knowing. Suppose furthermore that these die switches occur randomly. All you observe is the sequence of faces without knowing whether the fair or loaded die is in place: the identity of the die is hidden from you. Hidden Markov models (HMMs) account for exactly these kinds of situations involving hidden variables. In the die example we can use an HMM with two states,  $L$  (loaded) and  $F$  (fair), for the two dies. At any point in time the HMM is in one of the two *states* corresponding to the die that is in use. Associated with each of the two states are the probabilities for the different faces to come up. These *emission probabilities* are unknown and can be learned from observations (the distribution is uniform for the fair die and non-uniform for the loaded one). Transitions between states (dies) are governed by unknown *transitions probabilities* that model how likely the die holders switch dies at any time. For two states, transitions are modeled by an unknown 2 by 2 *transition matrix* that can also be learned from observations.

An HMM can produce observations by randomly choosing transitions (die switches) and *observations* or *emissions* (faces that come up) which results in an *observation sequence* and an underlying *hidden state sequence*. However, HMMs are so useful because they can be applied in reverse: given an observation sequence, we can estimate good parameters (emission and transition probabilities) for the underlying HMM as well as the underlying hidden state sequence, which we never directly observed.

In a classical HMM (applied to stimulus-response modeling) the time lag between the stimulus and the response is fixed and together stimulus and response probabilistically depend on some hidden (non-observed) variable with Markov dynamics.

In an MPH, spike and stimuli also probabilistically depend on some hidden variable, but rather than being paired at a fixed time lag, spike and stimulus pairing is dynamic, governed by a probabilistic process. Note that MPHs are different from factorial hidden Markov models which employ a distributed state representation but model a single (possibly multidimensional) observation sequence [64].

### Formal Definition of the MPH

In the following, we present a precise definition of the MPH architecture and its learning and inference algorithms.

We denote the stimulus sequence by  $X = x_1, x_2, \dots, x_T$  and the spiking response by  $R = r_1, r_2, \dots, r_U$  where  $T$  and  $U$  are their respective durations (typically  $T = U$ ).  $x_i \in \mathbb{R}^n$  are real vectors (e.g. sound spectrograms) and  $r_u \in \{0, \dots, B\}$  are integers (e.g., number of spikes, typically  $r_u \in \{0, 1\}$  in small time bins with zero or one spike). We denote a position in the combined stimulus-response alignment matrix as  $(t, u)$ , Fig. 2c. The model has three types of hidden states: X-states, which model only the stimulus, R-States which model only the response, and M-States which jointly model stimulus and response (Fig. 2c). We denote the sets of these states by  $X_S$ ,  $R_S$ , and  $M_S$ . Additionally we define the union of states  $Z_S = X_S \cup R_S \cup M_S$ . We denote sequences of hidden states by  $\Gamma = \gamma_1, \gamma_2, \dots, \gamma_H$  with  $\gamma_i \in Z_S$  and use the notation  $\gamma_{(t,u)}$  to refer to the hidden state occupied at sequence position  $(t, u)$ . Note that in general  $T \neq U \neq H$  because all of stimulus, response, and hidden state sequences may be of different length. The parameters of the MPH are defined in the following.

$A$ : Matrix of transition probabilities.  $A_{ij}$  denotes the probability of transiting from hidden state  $i$  to hidden state  $j \in Z_S$

$I_i$ : Initial probability of hidden state  $i \in Z_S$

$F_i$ : Final probability of hidden state  $i \in Z_S$

$b_i(x_t)$  for  $i \in X_S$ : Emission probability density of the stimulus  $x_t$  given hidden X-state  $i \in X_S$

$b_i(r_u)$  for  $i \in R_S$ : Discrete emission probability distribution of the response  $r_u$  given hidden R-state  $i \in R_S$

$b_i(x_t, r_u)$  for  $i \in M_S$ : Mixed discrete-continuous emission probability of stimulus-response pair  $(x_t, r_u)$  given hidden state  $i \in M_S$

As emission probability densities associated with X and M states we use multivariate Gaussians or mixtures of Gaussians, respectively:

$$b_i(x) = \sum_{k=1}^K c_{ik} N(x, \mu_{ik}, \Sigma_{ik}), i \in X_S,$$

where  $c_{ik}$  is the weight of the  $k^{\text{th}}$  mixture component,  $K$  denotes the total number of mixture components (which may vary for different hidden states but this freedom is not reflected in our notation), and  $\mu_{ik}$  and  $\Sigma_{ik}$  denote Gaussian mean and covariance matrix of the  $k^{\text{th}}$  mixture component. For M states we keep track of one such density for each possible value of  $r_u$  (distinct stimulus emission for each spiking state).

In the following, we define algorithms for inference in MPHs. Some of them are generalizations of well-known algorithms for normal HMMs. To infer the spiking response for a given stimulus, we derive new algorithms. In the following we denote conditional probabilities of the form  $P(\text{Data} | \text{Model Parameters})$  simply by  $P(\text{Data})$ , i.e., for readability we will omit the dependence on model parameters.

### Generalized Viterbi Algorithm

Assume that we have trained MPH model parameters on some data and now would like to apply the MPH to novel stimulus-response pairs. In a switching model (Fig. 2e), we would like to estimate the most likely hidden state sequence given the data to identify the switching events. In a flexible timing model (Fig. 2b) we would like to determine the optimal alignment between stimulus and response to estimate the jitter of individual spikes. In that latter case, the alignment consists of temporal offsets between stimulus and response on a moment-to-moment basis.

The generalized Viterbi algorithm for MPHs can be applied in both situations to efficiently compute the most likely hidden state sequence  $\Gamma^*$  for a given stimulus-response sequence  $(X, R)$ :

$$\Gamma^* = \arg\max_{\Gamma} P(\Gamma | X, R) = \arg\max_{\Gamma} P(\Gamma, X, R).$$

We apply an extension of the Viterbi algorithm for classical HMMs [38]. First, let  $v_j(t, u)$  be the probability of the most likely sequence that models the stimulus up to (and including) time  $t$ , the response up to time  $u$ , and that ends in hidden state  $j$ . Additionally, for any state  $j$  and sequence position  $(t, u)$ , we keep track of the most likely precursor state in  $\eta_j(t, u)$ .  $v_j(t, u)$  and  $\eta_j(t, u)$  can be computed recursively (Table 1).

A good way to visualize the generalized Viterbi algorithm is to think of it as filling up an  $T \times U \times |Z_S|$  alignment tensor (Fig. 2f). The final state of the most likely hidden state sequence is then given by  $\gamma_H^* = \arg\max_{j \in Z_S} v_j^*(T, U)$  and the complete state sequence can be obtained by iteratively back-tracking the most likely precursor states  $\eta_j(t, u)$ .

### Generalized Forward Algorithm

In many cases, we are interested in computing statistics over all possible sequences. For instance, to compute the probability

**Table 1.** The generalized Viterbi algorithm.

<b>Initialization:</b>	$v_j(0,0) = I_j$
<b>Recursion:</b>	$j \in M_S:$
	$v_j(t,u) = b_j(x_t, r_u) \max_{i \in Z_S} [A_{ij} v_i(t-1, u-1)]$
	$\eta_j(t,u) = \arg \max_{i \in Z_S} [A_{ij} v_i(t-1, u-1)]$
	$j \in X_S:$
	$v_j(t,u) = b_j(x_t) \max_{i \in Z_S} [A_{ij} v_i(t-1, u)]$
	$\eta_j(t,u) = \arg \max_{i \in Z_S} [A_{ij} v_i(t-1, u)]$
	$j \in Y_S:$
	$v_j(t,u) = b_j(r_u) \max_{i \in Z_S} [A_{ij} v_i(t, u-1)]$
	$\eta_j(t,u) = \arg \max_{i \in Z_S} [A_{ij} v_i(t, u-1)]$
<b>Termination:</b>	$v_j^*(T, U) = v_j(T, U) F_j$

doi:10.1371/journal.pcbi.1003508.t001

$P(X, R)$  of generating a sequence pair  $(X, R)$  given a particular MPH (for example to compare different MPHs), we need to consider the overall probability of  $(X, R)$  independent of the alignment. Hence we have to consider all possible hidden state sequences and not just the one with maximal likelihood. First, let  $\alpha_j(t, u)$  be the probability of observing the stimulus up to (and including) time  $t$ , the response up to time  $u$ , and of ending in hidden state  $j$ . The computation of  $\alpha_j(t, u)$  is very similar to the computation of  $v_j(t, u)$ , except that the max operation is replaced with a summation (to take all hidden state sequences into account, Table 2).

### Generalized Backward Algorithm

The backward algorithm is analogous to the forward algorithm. We present it here because it is an integral part of the EM algorithm for MPHs and the computation of posterior probabilities (see below). The backward probability  $\beta_j(t, u)$  is the probability of observing the stimulus from time  $t$  to the end and the response from time  $u$  to the end (excluding  $x_t, r_u$ ), beginning at position  $(t, u)$  and in hidden state  $j$  ( $\gamma_{(t,u)} = j$ ).  $\beta_j(t, u)$  is computed recursively (Table 3).

### Computing Posterior Probabilities

Assume we have trained an MPH on some data and want to determine the probability distribution over hidden states for a given stimulus-response pair and sequence position  $(t, u)$ . Building on the definitions of  $\alpha_j(t, u)$  and  $\beta_j(t, u)$  (Table 2 and 3), the posterior probability  $P(\gamma_{(t,u)} = j | X, R)$  of being in hidden state  $j$  at sequence position  $(t, u)$  given sequence-pair  $(X, R)$  can be

expressed in terms of forward and backward probabilities:

$$P(\gamma_{(t,u)} = j | X, R) = \frac{\alpha_j(t, u) \beta_j(t, u)}{P(X, R)}. \quad (8)$$

### Computing Alignment Kernels

Intuitively, the alignment kernel  $F_A(\tau)$  is a histogram of spike shifts over all possible state paths weighted by their respective probability.

$$F_A(\tau) \propto P(\tau = u - t, \gamma_{(t,u)} \in M_S, r_u = 1 | X, R)$$

The alignment kernel is easily computed using posterior probabilities (Eq. 8) in M- and R-States at all sequence positions  $(t, u)$  which fulfill  $r_u = 1$ :

$$F_A(\tau) \propto \sum_{u-t=\tau} P(\gamma_{(t,u)} = j | X, R). \\ j \in M_S \cup R_S \\ r_u = 1$$

Negative lags  $\tau$  in the alignment kernel imply that spikes occur before corresponding events in the stimulus, whereas positive lags imply that spikes occur thereafter.

**Table 2.** The generalized forward algorithm.

<b>Initialization:</b>	$\alpha_j(0,0) = I_j$
<b>Recursion:</b>	$j \in M_S:$
	$\alpha_j(t,u) = b_j(x_t, r_u) \sum_{i \in Z_S} [A_{ij} \alpha_i(t-1, u-1)]$
	$j \in X_S:$
	$\alpha_j(t,u) = b_j(x_t) \sum_{i \in Z_S} [A_{ij} \alpha_i(t-1, u)]$
	$j \in Y_S:$
	$\alpha_j(t,u) = b_j(r_u) \sum_{i \in Z_S} [A_{ij} \alpha_i(t, u-1)]$
<b>Termination:</b>	$P(X, R) = \sum_{j \in Z_S} \alpha_j(T, U) F_j$

doi:10.1371/journal.pcbi.1003508.t002

**Table 3.** The generalized backward algorithm.

<b>Initialization:</b>		$\beta_j(T, U) = F_j$
<b>Recursion:</b>	$j \in M_S$	$\beta_j(t, u) = \sum_{i \in Z_S} [A_{ji} b_j(x_{t+1}, r_{u+1}) \beta_i(t+1, u+1)]$
	$j \in X_S$	$\beta_j(t, u) = \sum_{i \in Z_S} [A_{ji} b_j(x_{t+1}) \beta_i(t+1, u)]$
	$j \in Y_S$	$\beta_j(t, u) = \sum_{i \in Z_S} [A_{ji} b_j(r_{u+1}) \beta_i(t, u+1)]$
<b>Termination:</b>		$P(X, R) = \sum_{j \in Z_S} \beta_j(0, 0) I_j$

doi:10.1371/journal.pcbi.1003508.t003

### Learning Model Parameters

To train an MPH on a set of stimulus-response pairs, we apply a generalization of the EM algorithm. That algorithm is analogous to its normal HMM counterpart [38,40]. In the expectation step, the  $\alpha_j(t, u)$  and  $\beta_j(t, u)$  (Table 2 and 3) are used to compute the probability of each state at each sequence position, as well as the expected number of transitions between hidden state pairs. The model parameters are then re-estimated in such a way as to locally maximize the likelihood of the stimulus-response pair. For simplicity of notation we define  $\zeta_{ij}(t, u)$  as the probability of transiting from state  $j$  to state  $i$  at sequence position  $(t, u)$ :

$$\zeta_{ij}(t, u) = \begin{cases} \alpha_i(t, u) A_{ij} b_j(t, u) \beta_j(t+1, u+1) / P(X, R) & \text{if } j \in M_S \\ \alpha_i(t, u) A_{ij} b_j(t) \beta_j(t+1, u) / P(X, R) & \text{if } j \in X_S \\ \alpha_i(t, u) A_{ij} b_j(u) \beta_j(t, u+1) / P(X, R) & \text{if } j \in Y_S \end{cases}$$

Based on  $\zeta_{ij}(t, u)$ , the new transition probabilities are given by

$$\hat{A}_{ij} = \frac{\sum_t \sum_u \zeta_{ij}(t, u)}{\sum_t \sum_u \sum_j \zeta_{ij}(t, u)}.$$

Initial probabilities are updated similarly:

$$\hat{I}'_i \propto \begin{cases} \sum_j \zeta_{ij}(1, 1) & \text{if } j \in M_S \\ \sum_j \zeta_{ij}(1, 0) & \text{if } j \in X_S \\ \sum_j \zeta_{ij}(0, 1) & \text{if } j \in Y_S \end{cases}.$$

The new discrete emission probabilities for R-States are given by

$$\hat{\beta}_i(r) = \frac{\sum_t \sum_u \mathbb{I}_{r_u=r} \sum_j \zeta_{ij}(t, u)}{\sum_t \sum_u \sum_j \zeta_{ij}(t, u)},$$

where  $i \in Y_S$  and  $r \in \{0, \dots, B\}$ .

The update of emission density parameters for X states depends on the type of continuous probability distribution used. For Gaussian mixtures with  $K$  mixture components, new means and covariance matrices for the components can be computed as follows. For simplicity, we first define  $\gamma_i^k(t, u)$ , where  $i \in X_S$  and

$k \in \{1..K\}$ :

$$\gamma_i^k(t, u) = \left( \frac{\alpha_i(t, u) \beta_i(t, u)}{\sum_j \alpha_j(t, u) \beta_j(t, u)} \right) \left( \frac{c_{ik} v(X_t, \mu_{ik}, \Sigma_{ik})}{\sum_l c_{il} v(X_t, \mu_{il}, \Sigma_{il})} \right)$$

The updated mixture weights,  $\hat{c}_{ik}$ , the means,  $\hat{\mu}_{ik}$ , and the covariance matrices  $\hat{\Sigma}_{ik}$  are then computed as follows:

$$\hat{c}_{ik} = \frac{\sum_t \sum_u \gamma_i^k(t, u)}{\sum_t \sum_u \sum_i \gamma_i^k(t, u)},$$

$$\hat{\mu}_{ik} = \frac{\sum_t (\sum_u \gamma_i^k(t, u)) X_t}{\sum_t \sum_u \gamma_i^k(t, u)},$$

$$\hat{\Sigma}_{ik} = \frac{\sum_t ((\sum_u \gamma_i^k(t, u)) [X_t - \hat{\mu}_{ik}][X_t - \hat{\mu}_{ik}]^T)}{\sum_t \sum_u \gamma_i^k(t, u)}.$$

The updates for M-states are analogous. To compute the updated parameters of the mixture associated with  $r \in \{0..B\}$  (where  $B+1$  is the number of possible neural responses, i.e.  $B$  is the maximum number of spikes per time bin), we sum only over those sequence positions  $u$  that fulfill  $r_u = r$ .

### Most Likely Pair of Response and Hidden State Sequences

Given an MPH that was trained on some stimulus-response pairs, we can predict spiking responses to novel stimuli. This is known as encoding. Conversely, we can reconstruct stimuli from spiking responses, known as decoding. In the following, we derive two encoding algorithms for MPHs. First, we show how to compute the most likely pair of hidden-state and neural response sequences,  $(\Gamma, R)^*$ , for a given stimulus  $X$ . This algorithm is an extension of the generalized Viterbi algorithm (Table 1). We only present the algorithm for encoding. By symmetry, a decoding algorithm can be derived analogously.

Let again  $v_j(t, u)$  be the probability of the most likely hidden state sequence that models the stimulus up to time  $t$  and the response up to time  $u$  and ends in state  $j$ . We want to compute a neural response  $R^* = r_1^*, r_2^*, \dots, r_U^*$  such that  $P(R^*, \Gamma^*[R^*, X])$  is maximized, where  $\Gamma^*[R^*, X]$  denotes the most likely state path for the sequence pair  $[R^*, X]$  (Table 1). This is accomplished by

always choosing the instantaneous neural response  $r_u^*$ ,  $u=1..U$  such that it maximizes the emission probability in the recursion equations (Table 4).

As in the generalized Viterbi algorithm (Table 1), we keep track of the most likely precursor states in  $\eta_j(t, u)$ . Additionally, we store the emissions that maximize the first factor on the right hand side of the recursion equations as  $r_j^*(t, u)$ . We recover the most likely pair of hidden state sequence and neural response by considering the  $r_j^*(t, u)$  associated with the most likely state at that position (we assume that  $U=T$ ; generalization to unknown  $U$  is possible, but irrelevant for our purposes).

This encoding strategy yields a spike train which depends on the most likely hidden state sequence. Such dependence can be a problem if many pairs of hidden state sequences exist with similarly high probability. Also, another caveat is that this algorithm does not provide spiking probabilities. Ideally, we would like to account for all possible hidden state sequences and compute an overall spiking or response probability for each point in time. Such improvement can be done through an extension of the forward algorithm, presented next.

### Computing the Response Probability Distribution as a Function of Time

Here we compute the probability distribution  $P(r_t|X)$  of the response  $r_t$  at time  $t$  given a stimulus sequence  $X$ . We can retrieve this probability as a posterior (using Eq. 8) after rewriting our model in the following way.

1. Replace each M-State by X- and R-states. If the response is encoded using the two symbols 0 and 1 ( $r_u \in \{0, 1\}$ ), an M-State is replaced by two X-states,  $X^0$  and  $X^1$ , representing  $P(x_t|r_u=0)$  and  $P(x_t|r_u=1)$  respectively and two R-states:  $R^0$  which never generates a spike, and  $R^1$  which always generates a spike.  $X^0$  is connected to  $R^0$  and  $X^1$  is connected to  $R^1$  (with probability 1 in both cases). Each connection onto the former M-state is now replaced by a pair of connections to  $X^0$  and  $X^1$ , with transition probabilities each given by the product of the original transition probability and the marginal probability of a non-spike ( $X^0$ ) or spike ( $X^1$ ) (computed by integrating the emission density of the M-state). By construction the model that results from applying this step is equivalent to the original model as far as inference is concerned.
2. Replace each of the R states in the model (except those that have been generated in step 1) by two R-states:  $R^0$  that never emits a spike and  $R^1$  that always emits a spike. As in Step 1, the

probability of spiking is encoded in the new transitions onto  $R^0$  and  $R^1$ . By construction, the resulting model is equivalent as far as inference is concerned.

With this reformulation, we can now easily express  $P(r_t|X)$  using sums over posterior probabilities of the  $R^0$  and  $R^1$  states:

$$P(r_t = 1 | X) = \frac{\sum_{u=1..U, j \in \{R^1\}} P_P(\gamma_{(t,u)} = j | X)}{\sum_{u=1..U, j \in \{R^1 \cup R^0\}} P_P(\gamma_{(t,u)} = j | X)},$$

where  $P_P(\gamma_{(t,u)} = j | X)$  denotes the posterior probability of hidden state  $j$  in the rewritten model,  $\{R^1\}$  and  $\{R^0\}$  denote the sets of all ‘spiking’ and non-spiking R-states, respectively. Note that by construction  $P_P(\gamma_{(t,u)} = j | X)$  is independent of the response  $R$ . In this paper we always use this algorithm for inferring spiking probabilities in MPHs.

**Cascaded MPHs.** Inspired by [26] and the standard practice of forming model cascades in neural response modeling [1,27,28], we cascade the MPH, forming an NNP (non-linear-non-linear) model. For the M-MPH (section on LNP equivalence), we can realize arbitrary LNP non-linearities  $h: \mathbb{R} \rightarrow \mathbb{R}$ . Given  $h$  we define a mapping  $z: z(p) = h(-\phi^{-1}(p) + c)$  operating on the posterior spike probability  $p$  in Eq. 4. Applying this mapping  $z$  to Eq. 4 yields the desired spike probability  $p_{spike} = z(p) = h(x^T \mu_1)$ .

Alternatively, we can estimate the optimal mapping  $z(p)$  that yields the nonlinearity  $h$  that best describes the data. We estimate this mapping from the data using the conditional probability

$$z(p) = P(r_t = 1 | p) = \frac{P(r_t = 1, p)}{P(p)}. \quad (9)$$

Thus, the optimal (discretized) mapping  $z$  corresponds to point-wise division of two histograms, the histogram of posterior spiking probabilities given an actual spike in the numerator and the histogram of all posterior spiking probabilities in the denominator (see also [1]).

In practice, we first estimate the MPH parameters and then re-estimate the non-linearity via the mapping  $z$  in (Eq. 9). When applying this cascaded MPH, we first compute the posterior spiking probabilities and then remap these using  $z$ . These response predictions are bound to give better results on the training set and will also improve validation performance (unless the mapping  $z$  is over-fitted).

**Table 4.** Extended Viterbi algorithm to compute most likely pair of hidden state and neural response sequences for a given stimulus.

<b>Initialization:</b>	$v_j(0,0) = I_j$
<b>Recursion:</b> $j \in M_S$ :	$v_j(t,u) = \max_{r_u} [b_j(x_t, r_u)] \max_{i \in Z_S} [A_{ij} v_i(t-1, u-1)]$
	$r_j^*(t,u) = \max_{r_u} [b_j(x_t, r_u)]$
$j \in X_S$ :	$v_j(t,u) = b_j(x_t) \max_{i \in Z_S} [A_{ij} v_i(t-1, u)]$
$j \in Y_S$ :	$v_j(t,u) = \max_{r_u} b_j(r_u) \max_{i \in Z_S} [A_{ij} v_i(t, u-1)]$
	$r_j^*(t,u) = \max_{r_u} [b_j(r_u)]$
<b>Termination:</b>	$v_j^*(T, U) = v_j(T, U) F_j$

doi:10.1371/journal.pcbi.1003508.t004



## Computational Complexity and Optimization of the Algorithms

Filling out the alignment tensor used to compute forward and backward probabilities (Fig. 2f) in a fully connected model requires  $O(TUS^2)$  computations and additional  $O(TS)$  computations for emission probabilities in M and X states (as before,  $T$  denotes the length of the  $X$  sequence (stimulus),  $U$  the length of the  $R$  sequence (response); and  $S$  is the number of hidden states). We usually reduce this complexity by limiting the allowed temporal offset between stimulus and response to a maximal lag set by a parameter  $w$ . In that case, we compute only the part of the alignment tensor within a band of width  $w$  around the diagonal. Hence, the complexity reduces to  $O(TwS^2)$ . In the EM algorithm, the computational complexity is  $O(TwS^2)$ .

The MPHs we studied had mostly constrained parameters, in particular constrained covariance matrices and means. We have found that free covariance matrices tend to make the models prone to over fitting and slow down training as more iterations of the EM algorithm are required (for instance, the M-MPH discussed in the section on LNP equivalence reaches the optimum in one iteration. Using free covariance matrices, convergence is gradual and it takes many more steps for the likelihood change to drop below a predefined threshold).

The EM algorithm only converges to local optima; we found that this problem can be alleviated by running the training several times from different initializations (compare Fig. 5D and the accompanying text).

## References

- Schwartz O, Pillow JW, Rust NC, Simoncelli EP (2006) Spike-triggered neural characterization. *J Vis* 6: 484–507. Available: <http://www.ncbi.nlm.nih.gov/pubmed/16889482>. Accessed 17 February 2014.
- Sharpee T, Rust NC, Bialek W (2004) Analyzing neural responses to natural signals: maximally informative dimensions. *Neural Comput* 16: 223–250. Available: <http://www.ncbi.nlm.nih.gov/pubmed/15006095>. Accessed 17 February 2014.
- Benda J, Golisch T, Machens CK, Herz AV (2007) From response to stimulus: adaptive sampling in sensory physiology. *Curr Opin Neurobiol* 17: 430–436. Available: <http://dx.doi.org/10.1016/j.conb.2007.07.009>. Accessed 21 July 2011.
- Paninski L (2003) Convergence properties of three spike-triggered analysis techniques. *Network* 14: 437–464. Available: <http://www.ncbi.nlm.nih.gov/pubmed/12938766>.
- Park IM, Archer EW, Priebe N, Pillow JW (2013) Spectral methods for neural characterization using generalized quadratic models. *Advances in Neural Information Processing Systems*, pp. 2454–2462. Available: <http://papers.nips.cc/paper/4993-spectral-methods-for-neural-characterization-using-generalized-quadratic-models>. Accessed 22 January 2014.
- Steveninck RDR V, Bialek W (1988) Real-Time Performance of a Movement-Sensitive Neuron in the Blowfly Visual System: Coding and Information Transfer in Short Spike Sequences. *Proc R Soc B Biol Sci* 234: 379–414. Available: <http://rspb.royalsocietypublishing.org/content/234/1277/379.short>. Accessed 7 July 2013.
- Golisch T, Meister M (2008) Rapid neural coding in the retina with relative spike latencies. *Science* 319: 1108–1111. Available: <http://www.sciencemag.org/content/319/5866/1108.abstract>. Accessed 4 July 2011.
- Tolhurst DJ, Smyth D, Thompson ID (2009) The sparseness of neuronal responses in ferret primary visual cortex. *J Neurosci* 29: 2355–2370. Available: <http://www.jneurosci.org/cgi/content/abstract/29/8/2355>. Accessed 5 July 2011.
- Levick WR (1973) Variation in the response latency of cat retinal ganglion cells. *Vision Res* 13.
- Gawne TJ (2000) The simultaneous coding of orientation and contrast in the responses of V1 complex cells. *Exp Brain Res* 133: 293–302. Available: <http://www.ncbi.nlm.nih.gov/pubmed/10958519>. Accessed 8 July 2011.
- Reich DS, Mechler F, Victor JD (2001) Temporal Coding of Contrast in Primary Visual Cortex: When, What, and Why. *J Neurophysiol* 85: 1039–1050. Available: <http://jn.physiology.org/cgi/content/abstract/85/3/1039>. Accessed 6 July 2011.
- Gawne TJ, Kjaer TW, Richmond BJ (1996) Latency: another potential code for feature binding in striate cortex. *J Neurophysiol* 76: 1356–1360. Available: <http://jn.physiology.org/cgi/content/abstract/76/2/1356>. Accessed 6 July 2011.
- Oram MW, Xiao D, Drietschel B, Payne KR (2002) The temporal resolution of neural codes: does response latency have a unique role? *Philos Trans R Soc Lond B Biol Sci* 357: 987–1001. Available: <http://rspb.royalsocietypublishing.org/cgi/content/abstract/357/1424/987>. Accessed 22 July 2010.
- Kiani R, Esteky H, Tanaka K (2005) Differences in onset latency of macaque inferotemporal neural responses to primate and non-primate faces. *J Neurophysiol* 94: 1587–1596. Available: <http://jn.physiology.org/cgi/content/abstract/94/2/1587>. Accessed 7 September 2010.
- Dimitrov AG, Sheiko MA, Baker J, Yen S-C (2009) Spatial and temporal jitter distort estimated functional properties of visual sensory neurons. *J Comput Neurosci* 27: 309–319. Available: <http://www.springerlink.com/content/c1365573t6285360/>. Accessed 12 April 2012.
- Golisch T (2006) Estimating receptive fields in the presence of spike-time jitter. *Network* 17: 103–129. Available: <http://informahealthcare.com/doi/abs/10.1080/09548980600569670>. Accessed 12 April 2012.
- Buonomano D V, Maass W (2009) State-dependent computations: spatiotemporal processing in cortical networks. *Nat Rev Neurosci* 10: 113–125. Available: <http://www.nature.com/nrn/journal/v10/n2/full/nrn2558.html#B12>. Accessed 6 July 2011.
- Destexhe A, Contreras D (2006) Neuronal computations with stochastic network states. *Science* 314: 85–90. Available: <http://www.sciencemag.org/content/314/5796/85.abstract>. Accessed 21 July 2010.
- Sachdev RNS, Ebner FF, Wilson CJ (2004) Effect of subthreshold up and down states on the whisker-evoked response in somatosensory cortex. *J Neurophysiol* 92: 3511–3521. Available: <http://jn.physiology.org/cgi/content/abstract/92/6/3511>. Accessed 18 July 2012.
- Bartlett EL, Wang X (2005) Long-lasting modulation by stimulus context in primate auditory cortex. *J Neurophysiol* 94: 83–104. Available: <http://jn.physiology.org/cgi/content/abstract/94/1/83>. Accessed 20 June 2011.
- Geffen MN, De Vries SEJ, Meister M (2007) Retinal Ganglion Cells Can Rapidly Change Polarity from Off to On. *PLoS Biol* 5: 11. Available: <http://www.ncbi.nlm.nih.gov/pubmed/17341132>.
- Keller GB, Hahnloser RHR (2009) Neural processing of auditory feedback during vocal practice in a songbird. *Nature* 457: 187–190. Available: <http://dx.doi.org/10.1038/nature07467>. Accessed 17 June 2011.
- Schmidt MF, Konishi M (1998) Gating of auditory responses in the vocal control system of awake songbirds. *Nat Neurosci* 1: 513–518. Available: <http://www.ncbi.nlm.nih.gov/pubmed/10196550>. Accessed 18 September 2012.
- Kouh M, Sharpee TO (2009) Estimating linear–nonlinear models using Rényi divergences. *Network* 20: 49–68. Available: <http://informahealthcare.com/doi/abs/10.1080/09548980902950891>. Accessed 12 April 2012.
- Sen K, Theunissen FE, Doupe a J (2001) Feature analysis of natural sounds in the songbird auditory forebrain. *J Neurophysiol* 86: 1445–1458. Available: <http://www.ncbi.nlm.nih.gov/pubmed/11535690>.

## Subjects and Electrophysiology

All experiments were carried out in accordance with protocols approved by the Veterinary Office of the Canton of Zurich, Switzerland. Data were collected from juvenile male zebra finches (60–92 days old). The electrophysiological procedures are explained in detail elsewhere [65]. Briefly, microdrives were implanted using methods previously described [65]. After each experiment, the brain was removed for histological examination of unstained slices to verify the location of reference lesions. Cells were recorded during singing. During recording sessions, birds were housed in a sound isolation chamber equipped with a microphone. Extracellular voltage traces were digitized at 33 kHz and recorded for offline spike sorting.

## Acknowledgments

We thank Sylvia Schröder for very helpful discussions of the manuscript. We also thank Alexei Vyssotski and Georg Keller for providing the recorded spike trains in singing birds. Additionally, we thank Michael Pfeiffer and Florian Blättler for helpful discussions.

## Author Contributions

Conceived and designed the experiments: SK RHRH. Analyzed the data: SK. Contributed reagents/materials/analysis tools: SK. Wrote the paper: SK RHRH.

26. Hastie T, Tibshirani R, Friedman J (2009) The Elements of Statistical Learning. Elements 1: 337–387. Available: <http://www.springerlink.com/index/10.1007/b94608>.
27. Theis L, Chagas AM, Arnstein D, Schwarz C, Bethge M (2013) Beyond GLMs: A Generative Mixture Modeling Approach to Neural System Identification. *PLoS Comput Biol* 9: e1003356. Available: <http://www.ploscompbiol.org/article/info:doi/10.1371/journal.pcbi.1003356#pcbi.1003356.e057>. Accessed 18 December 2013.
28. Pillow JW, Simoncelli EP (2006) Dimensionality reduction in neural models: an information-theoretic generalization of spike-triggered average and covariance analysis. *J Vis* 6: 414–428. Available: <http://www.ncbi.nlm.nih.gov/pubmed/16889478>.
29. Movshon JA, Thompson ID, Tolhurst DJ (1978) Receptive field organization of complex cells in the cat's striate cortex. *J Physiol* 283: 79–99. Available: [http://jp.physoc.org/content/283/1/79.abstract?ikey=17f32190a8f456a0d4124f1439738b91a9d161ff&keytype2=tf\\_ipscsha](http://jp.physoc.org/content/283/1/79.abstract?ikey=17f32190a8f456a0d4124f1439738b91a9d161ff&keytype2=tf_ipscsha). Accessed 17 December 2012.
30. Adelson EH, Bergen JR (1985) Spatiotemporal energy models for the perception of motion. *J Opt Soc Am A* 2: 284–299. Available: <http://www.ncbi.nlm.nih.gov/pubmed/3973762>. Accessed 2 December 2012.
31. Touryan J, Felsen G, Dan Y (2005) Spatial structure of complex cell receptive fields measured with natural images. *Neuron* 45: 781–791. Available: <http://www.ncbi.nlm.nih.gov/pubmed/15748852>. Accessed 17 June 2011.
32. Naie K, Hahnloser RHR (2011) Regulation of learned vocal behavior by an auditory motor cortical nucleus in juvenile zebra finches. *J Neurophysiol* 106: 291–300. doi:10.1152/jn.01035.2010.
33. Cardin JA, Raksit JN, Schmidt MF (2005) Sensorimotor nucleus Nif is necessary for auditory processing but not vocal motor output in the avian song system. *J Neurophysiol* 93: 2157–2166. Available: <http://jn.physiology.org/content/93/4/2157.short>. Accessed 27 July 2012.
34. Hosino T, Okanoya K (2000) Lesion of a higher-order song nucleus disrupts phrase level complexity in Bengalese finches. *Neuroreport* 11: 2091–2095. Available: <http://www.ncbi.nlm.nih.gov/pubmed/10923650>.
35. Okanoya K (2004) The Bengalese finch: a window on the behavioral neurobiology of birdsong syntax. *Ann N Y Acad Sci* 1016: 724–735. Available: <http://www.ncbi.nlm.nih.gov/pubmed/15313802>. Accessed 22 August 2012.
36. Lewandowski BC, Schmidt M (2011) Short bouts of vocalization induce long-lasting fast  $\gamma$  oscillations in a sensorimotor nucleus. *J Neurosci* 31: 13936–13948. Available: <http://www.jneurosci.org/content/31/39/13936.short>. Accessed 25 September 2012.
37. McCasland J (1987) Neuronal control of bird song production. *J Neurosci* 7: 23–39. Available: <http://www.jneurosci.org/content/7/1/23.short>. Accessed 25 September 2012.
38. Durbin R, Eddy S, Krogh A, Mitchison G (1999) Biological Sequence Analysis. Cambridge: Cambridge University Press.
39. Singh NC, Theunissen FE (2003) Modulation spectra of natural sounds and ethological theories of auditory processing. *J Acoust Soc Am* 114: 3394. Available: <http://link.aip.org/link/JASMAN/114/3394/1>. Accessed 21 May 2013.
40. Rabiner LR (1989) A tutorial on hidden Markov models and selected applications in speech recognition. *Proc IEEE* 77: 257–286. Available: [http://ieeexplore.ieee.org/xpl/freeabs\\_all.jsp?arnumber=18626](http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=18626). Accessed 27 July 2011.
41. Aldworth ZN, Miller JP, Gedeon T, Cummins GI, Dimitrov AG (2005) Dejittered spike-conditioned stimulus waveforms yield improved estimates of neuronal feature selectivity and spike-timing precision of sensory interneurons. *J Neurosci* 25: 5323–5332. Available: <http://www.ncbi.nlm.nih.gov/pubmed/15930380>.
42. Gat I, Tishby N, Abeles M (1997) Hidden Markov modelling of simultaneously recorded cells in the associative cortex of behaving monkeys. *Network: Comput. Neural Syst* 8: 297–322. Available: [http://informahealthcare.com/doi/abs/10.1088/0954-898X\\_8\\_3\\_005](http://informahealthcare.com/doi/abs/10.1088/0954-898X_8_3_005). Accessed 20 November 2012.
43. Abeles M, Bergman H, Gat I, Meilijson I, Seidemann E, et al. (1995) Cortical activity flips among quasi-stationary states. *Proc Natl Acad Sci U S A* 92: 8616–8620. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=41017&tool=pmcentrez&rendertype=abstract>. Accessed 4 July 2011.
44. Camproux AC, Saunier F, Chouvet G, Thalabard JC, Thomas G (1996) A hidden Markov model approach to neuron firing patterns. *Biophys J* 71: 2404–2412. Available: <http://linkinghub.elsevier.com/retrieve/pii/S0006349596794341>.
45. Radons G, Becker JD, Dülfer B, Krüger J (1994) Analysis, classification, and coding of multielectrode spike trains with hidden Markov models. *Biol Cybern* 71: 359–373. Available: <http://www.ncbi.nlm.nih.gov/pubmed/7948227>.
46. Rainer G, Miller EK (2000) Neural ensemble states in prefrontal cortex identified using a hidden Markov model with a modified EM algorithm. *Neurocomputing* 32–33: 961–966. Available: [http://dx.doi.org/10.1016/S0925-2312\(00\)00266-6](http://dx.doi.org/10.1016/S0925-2312(00)00266-6). Accessed 20 November 2012.
47. Weber AP, Hahnloser RHR (2007) Spike correlations in a songbird agree with a simple markov population model. *PLoS Comput Biol* 3: e249. Available: <http://dx.plos.org/10.1371/journal.pcbi.0030249>. Accessed 20 November 2012.
48. Dan G (2006) Efficient estimation of hidden state dynamics from spike trains. *Adv Neural Inf Process Syst* 18: 227–234. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.67.7313&rep=rep1&type=pdf>.
49. Ghahramani Z, Hinton GE (2000) Variational Learning for Switching State-Space Models. *Neural Comput* 12: 831–864. Available: <http://www.mitpressjournals.org/doi/abs/10.1162/089976600300015619>. Accessed 20 April 2012.
50. Wu W, Kulkarni JE, Hatsopoulos NG, Paninski L (2009) Neural Decoding of Hand Motion Using a Linear State-Space Model With Hidden States. *IEEE Trans Neural Syst Rehabil Eng* 17: 370–378. Available: <http://www.ncbi.nlm.nih.gov/pubmed/19497822>.
51. Lawhern V, Wu W, Hatsopoulos N, Paninski L (2010) Population decoding of motor cortical activity using a generalized linear model with hidden states. *J Neurosci Methods* 189: 267–280. Available: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2921213&tool=pmcentrez&rendertype=abstract>.
52. Ahrens MB, Linden JF, Sahani M (2008) Nonlinearities and contextual influences in auditory cortical responses modeled with multilinear spectrotemporal methods. *J Neurosci* 28: 1929–1942. Available: <http://www.jneurosci.org/content/28/8/1929.short>. Accessed 22 May 2013.
53. Rust NC, Schwartz O, Movshon JA, Simoncelli EP (2005) Spatiotemporal elements of macaque v1 receptive fields. *Neuron* 46: 945–956. Available: [http://www.cell.com/neuron/fulltext/S0896-6273\(05\)00468-X](http://www.cell.com/neuron/fulltext/S0896-6273(05)00468-X). Accessed 21 May 2013.
54. Touryan J, Lau B, Dan Y (2002) Isolation of Relevant Visual Features from Random Stimuli for Cortical Complex Cells. *J Neurosci* 22: 10811–10818. Available: [http://www.jneurosci.org/content/22/24/10811.abstract?ikey=75844c6092d5d53a05a047a1aa8a909c67691529&keytype2=tf\\_ipscsha](http://www.jneurosci.org/content/22/24/10811.abstract?ikey=75844c6092d5d53a05a047a1aa8a909c67691529&keytype2=tf_ipscsha). Accessed 17 December 2012.
55. Eickenberg M, Rowekamp RJ, Kouh M, Sharpee TO (2012) Characterizing Responses of Translation-Invariant Neurons to Natural Stimuli: Maximally Informative Invariant Dimensions. *Neural Comput* 24: 2384–23421. Available: [http://www.mitpressjournals.org/doi/abs/10.1162/NECO\\_a\\_00330?journalCode=neco](http://www.mitpressjournals.org/doi/abs/10.1162/NECO_a_00330?journalCode=neco). Accessed 7 July 2013.
56. Bienenstock E (1995) A model of neocortex. *Network: Comput Neural Syst* 6: 179–224. Available: [http://informahealthcare.com/doi/abs/10.1088/0954-898X\\_6\\_2\\_004](http://informahealthcare.com/doi/abs/10.1088/0954-898X_6_2_004). Accessed 3 November 2011.
57. Ikegaya Y, Aaron G, Cossart R, Aronov D, Lampl I, et al. (2004) Synfire chains and cortical songs: temporal modules of cortical activity. *Science* 304: 559–564. Available: <http://www.sciencemag.org/content/304/5670/559.abstract>. Accessed 5 July 2011.
58. Abeles M, Hayon G, Lehmann D (2004) Modeling compositionality by dynamic binding of synfire chains. *J Comput Neurosci* 17: 179–201. Available: <http://www.springerlink.com/content/t565584112517216/>. Accessed 15 July 2011.
59. Blaettler F, Kollmorgen S, Herbst J, Hahnloser R (2011) Hidden Markov Models in the Neurosciences. In: Dymarski P, editor. *Hidden Markov Models, Theory And Applications*. InTech. pp. 169–186.
60. Victor JD, Purpura KP (1998) Metric-space analysis of spike trains: theory, algorithms, and application. *Netw Comput Neural Syst* 8: 127–164. Available: <http://arxiv.org/abs/q-bio/0309031>.
61. Ephraim Y, Dembo A, Rabiner L (1987) A minimum discrimination information approach for hidden Markov modeling. *ICASSP '87. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Institute of Electrical and Electronics Engineers, Vol. 12. pp. 25–28. Available: <http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=1169727&contentType=Conference+Publications>. Accessed 25 September 2012.
62. Fine S, Singer Y, Tishby N (1998) The Hierarchical Hidden Markov Model: Analysis and Applications. *Computer (Long Beach Calif)* 32: 41–62. Available: <http://www.springerlink.com/index/H7630R4U78J0XHU1.pdf>.
63. Stolcke A, Omohundro S (1993) Hidden Markov Model Induction by Bayesian Model Merging. *Adv Neural Inf Process Syst* 5: 11–18. Available: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.43.470&rep=rep1&type=pdf>.
64. Ghahramani Z, Jordan MI (1997) Factorial Hidden Markov Models. *Mach Learn* 29: 245–273. Available: <http://www.ncbi.nlm.nih.gov/pubmed/16204097>.
65. Fee MS, Leonardo A (2001) Miniature motorized microdrive and commutator system for chronic neural recording in small animals. *J Neurosci Methods* 112: 83–94. Available: <http://www.ncbi.nlm.nih.gov/pubmed/11716944>. Accessed 22 January 2013.